

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-233196

(43)Date of publication of application : 10.09.1993

(51)Int.Cl.

G06F 3/14

(21)Application number : 04-288571

(71)Applicant : INTERNATL BUSINESS MACH
CORP <IBM>

(22)Date of filing : 27.10.1992

(72)Inventor : BATES CARY L
RYAN JEFFREY M

(30)Priority

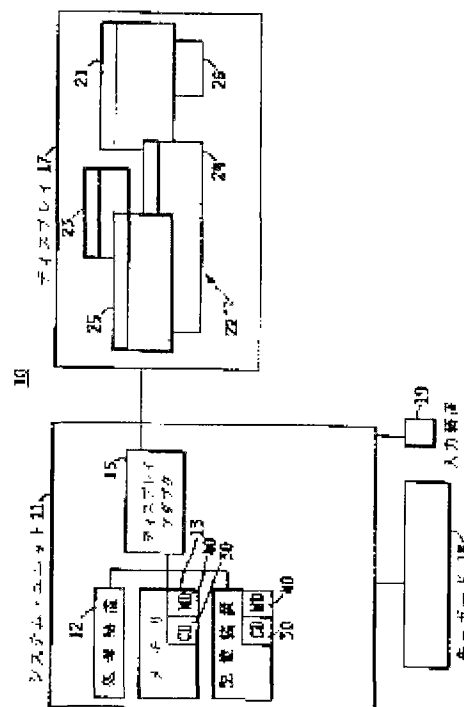
Priority number : 91 811548 Priority date : 20.12.1991 Priority country : US

(54) METHOD AND DEVICE FOR EFFICIENTLY DISPLAYING PLURAL WINDOWS ON DISPLAY SCREEN

(57)Abstract:

PURPOSE: To provide a method and device for distinctively and proportionally displaying a window on the display screen of a computer.

CONSTITUTION: The length of time when each of windows 21 to 26 displayed on a display screen 17 is in an active state (known as 'in-focus state') is monitored. At the time of receiving a command from a user, a window having been in a longer active state is more distinctively displayed from windows having been in a shorter active state. Namely each of the windows 21 to 26 is displayed on the display screen 17 with a window size in proportion to the length of the time having been in an active state.



(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平5-233196

(43)公開日 平成 5 年(1993) 9 月10日

(51)Int.Cl.⁵

G 0 6 F 3/14

識別記号

庁内整理番号

F I

技術表示箇所

3 5 0 A 7165-5B

審査請求 有 請求項の数13(全 37 頁)

(21)出願番号 特願平4-288571

(22)出願日 平成 4 年(1992)10月27日

(31)優先権主張番号 8 1 1 5 4 8

(32)優先日 1991年12月20日

(33)優先権主張国 米国 (U S)

(71)出願人 390009531

インターナショナル・ビジネス・マシー
ズ・コーポレーション

INTERNATIONAL BUSIN
ESS MACHINES CORPO
RATION

アメリカ合衆国10504、ニューヨーク州
アーモンク (番地なし)

(72)発明者 ケアリ・リー・ベイツ

アメリカ合衆国、ミネソタ州ロチェスタ、
ノウルレイン 3032

(74)代理人 弁理士 頼宮 孝一 (外 4 名)

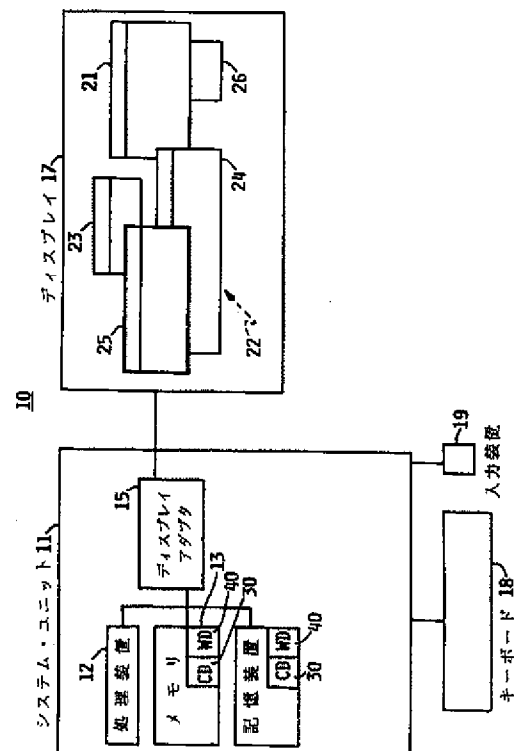
最終頁に続く

(54)【発明の名称】 ディスプレイ・スクリーン上に複数のウィンドウを能率的に表示するための方法及び装置

(57)【要約】 (修正有)

【目的】 コンピュータのディスプレイ・スクリーン上にウィンドウを区別的及び比例的に表示するための方法及び装置を提供する。

【構成】 ディスプレイ・スクリーン17上に表示されるウィンドウ21~26のそれぞれが活動状態(「イン・フォーカス状態」としても知られている)にある時間の長さがモニタされる。ユーザからのコマンドを受け取った時、より長い時間活動状態にあったウィンドウは、より短い時間活動状態にあったウィンドウよりもより区別的に表示される。すなわち、ウィンドウ21~26は、ウィンドウのそれぞれが活動状態にあった時間の長さに比例するウィンドウ・サイズを持ってディスプレイ・スクリーン17上に表示される。



【特許請求の範囲】

【請求項1】 ディスプレイ・スクリーン上に複数のウィンドウを能率的に表示するための方法であって、上記複数のウィンドウのそれぞれが活動状態にあった時間の長さをモニタするステップと、第Nのウィンドウが最小しきい値を越えるに十分に長い時間活動状態になかったことを決定するステップと、第1のウィンドウから第(N-1)のウィンドウを活動の降順に区別的に表示するステップと、第Nのウィンドウをアイコンとして表示するステップとを有する上記方法。

【請求項2】 ディスプレイ・スクリーン上に複数のウィンドウを能率的に表示するための方法であって、上記複数のウィンドウのそれぞれが活動状態にあった時間の長さをモニタするステップと、上記複数のウィンドウのそれぞれを活動状態にある時間の長さに比例するサイズで表示するステップとを有する上記方法。

【請求項3】 上記表示するステップは、上記ディスプレイ・スクリーン上でタイル表示領域を決定するステップと、上記タイル表示領域をH1領域とH2領域とに分割するステップと、上記H1領域をQ1領域とQ2領域とに分割するステップと、上記H2領域をQ3領域とQ4領域とに分割するステップとをさらに有する請求項2記載の方法。

【請求項4】 上記タイル表示領域を分割するステップは、上記第1のウィンドウから上記第Nのウィンドウが活動状態にあった時間の長さに対する上記第1のウィンドウ及び上記第2のウィンドウが活動状態にあった時間の長さに上記H1領域のサイズが比例し、上記第1のウィンドウから上記第Nのウィンドウが活動状態にあった時間の長さに対する上記第3のウィンドウから上記第Nのウィンドウが活動状態にあった時間の長さに上記H2領域のサイズが比例するように実行される請求項3記載の方法。

【請求項5】 上記H1領域を分割するステップは、上記第1のウィンドウ及び上記第2のウィンドウが活動状態にあった時間の長さに対する上記第1のウィンドウが活動状態にあった時間の長さに上記Q1領域のサイズが比例し、上記第1のウィンドウ及び上記第2のウィンドウが活動状態にあった時間の長さに対する上記第2のウィンドウが活動状態にあった時間の長さに上記Q2領域のサイズが比例するように実行される請求項4記載の方法。

【請求項6】 上記H2領域を分割するステップは、上記第3のウィンドウから上記第Nのウィンドウが活動状態にあった時間の長さに対する上記第3のウィンドウが活動状態にあった時間の長さに上記Q3領域のサイズが

比例し、上記第3のウィンドウから上記第Nのウィンドウが活動状態にあった時間の長さに対する上記第4のウィンドウから上記第Nのウィンドウが活動状態にあった時間の長さに上記Q4領域のサイズが比例するように実行される請求項5記載の方法。

【請求項7】 上記Q1領域に上記第1のウィンドウを表示するステップと、上記Q2領域に第2のウィンドウを表示するステップと、上記Q3領域に第3のウィンドウを表示するステップとをさらに有する請求項6記載の方法。

【請求項8】 上記第4のウィンドウから上記第Nのウィンドウがイン・フォーカス状態にあった時間の長さに対する上記第4のウィンドウがイン・フォーカス状態にあった時間の長さにQ4-Q1領域のサイズが比例し、上記第4のウィンドウから上記第Nのウィンドウがイン・フォーカス状態にあった時間の長さに対する上記第5のウィンドウがイン・フォーカス状態にあった時間の長さにQ4-Q2領域のサイズが比例するように、上記Q4領域を上記Q4-Q1領域と上記Q4-Q2領域とに分割するステップであってN=5であるものと、上記Q4-Q1領域に上記第4のウィンドウを表示するステップと、上記Q4-Q2領域に上記第5のウィンドウを表示するステップとをさらに有する請求項7記載の方法。

【請求項9】 ディスプレイ・スクリーン上に複数のウィンドウを能率的に表示するための方法であって、上記複数のウィンドウのそれぞれが活動状態にあった時間の長さをモニタするステップと、上記ディスプレイ・スクリーン上でタイル表示領域を決定するステップと、上記タイル表示領域をH1領域とH2領域とに等分割するステップと、上記H1領域をQ1領域とQ2領域とに等分割するステップと、上記H2領域をQ3領域とQ4領域とに等分割するステップと、上記Q1領域に上記第1のウィンドウを表示するステップと、上記Q2領域に上記第2のウィンドウを表示するステップと、上記Q3領域に上記第3のウィンドウを表示するステップとを有する上記方法。

【請求項10】 上記Q4領域をQ4-Q1領域とQ4-Q2領域とに等分割するステップと、上記Q4-Q1領域に上記第4のウィンドウを表示するステップと、上記Q4-Q2領域に上記第5のウィンドウを表示するステップとをさらに有する請求項9記載の方法。

【請求項11】 第6のウィンドウが最小しきい値を越

えるに十分に長い時間活動状態になかったことを決定するステップと、

上記第6のウィンドウをアイコンとして表示するステップとをさらに有する請求項10記載の方法。

【請求項12】 ディスプレイ・スクリーン上に複数のウィンドウを能率的に表示するためのコンピュータ・システムであって、

処理装置と上記処理装置に接続されたメモリと上記処理装置に接続された記憶装置と上記処理装置に接続されたディスプレイ・アダプタとを有するシステム・ユニットと、

上記システム・ユニットに接続されたディスプレイと、上記複数のウィンドウのそれぞれが活動状態にあった時間の長さをモニタするための、上記処理装置における手段と、

上記ディスプレイ上に上記複数のウィンドウのそれぞれを活動状態にある時間の長さに比例するサイズで表示するための、上記処理装置における手段とを有する上記コンピュータ・システム。

【請求項13】 ディスプレイ・スクリーン上に複数のウィンドウを能率的に表示するためのプログラム・プロダクトであって、

上記複数のウィンドウのそれぞれが活動状態にあった時間の長さをモニタするための手段と、

上記複数のウィンドウのそれぞれを活動状態にある時間の長さに比例するサイズで表示するための手段とを有する上記プログラム・プロダクト。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は、データ処理分野に関する。より詳細には、この発明は、コンピュータのディスプレイ・スクリーン上でのウィンドウの比例表示に関する。

【0002】

【従来の技術】 アップル・コンピュータ社により最初に市販され、後にマイクロソフト社によりその「Windows」プログラムに採用され、IBM社によりOS/2及びPresentation Manager（プレゼンテーション・マネージャ）に採用された「グラフィカル・ユーザ・インターフェース」として知られているものを使用するコンピュータ・システムは、かなり最近の従来技術である。これらのグラフィカル・ユーザ・インターフェース・システムの一つの共通の特徴は、コンピュータのディスプレイ・スクリーン上に多数のウィンドウまたは表示窓を同時に存在させることができることである。コンピュータのディスプレイ・スクリーン上に表示されたウィンドウのそれぞれにおいて、異なるアプリケーション・プログラムを同時に実行する（またはユーザからの入力を待つ）ことができる。また、単一のアプリケーション・プログラムは多数の異なるウィンドウを生成することができる。

ユーザは、異なるウィンドウ間であちこちに移動を行うためにマウスまたは他の入力装置を使用することができ、それによって多数の異なるタスクを実行する。

【0003】 これらのグラフィカル・ユーザ・インターフェース・システムは、DOS（一度に一つのアプリケーション・プログラムだけを実行及び表示することができる）のようなより一般的なオペレーティング・システムを越える多くの利点を提供するが、この付加的な機能はユーザに対して新たな問題を発生させた。グラフィカル・ユーザ・インターフェース・システムは、コンピュータのスクリーン上にほぼ無制限の数のウィンドウを表示する能力を提供するが、これはこれらのほぼ無制限の数のウィンドウを、ユーザがそれらを見ることができるようコンピュータのスクリーン上に表示することができることを意味しない。事実、これらのウィンドウの大多数が他のウィンドウによって部分的または完全に覆われてしまうことが十分にありうる。この問題は二つのウィンドウの場合に起こりうるが、これよりも多数のウィンドウが使用される時にはより顕著となる。

【0004】 ウィンドウのいくつかが部分的または完全に覆われた時には、ユーザは非常に多くの努力をすることなくそれらの多くを見つけることができないため、ユーザが種々のウィンドウ間であちこちにうまく移動を行うことは非常に難しくなる。グラフィカル・ユーザ・インターフェース・システムにおけるこの制限は、実際に同時に使用し、表示し、またはウィンドウによって同時に表示することができる異なるタスクまたはアプリケーション・プログラムの数を厳しく制限することによってそのようなシステムを得るという本当の目的をだいにしにしていまいがちである。

【0005】

【発明が解決しようとする課題】 この発明の主要な目的は、グラフィカル・ユーザ・インターフェース・システムの性能を高めることにある。

【0006】 この発明の他の目的は、ユーザが部分的または完全に覆われたウィンドウを見つけるためのより能率的な方法を提供することにある。

【0007】 この発明の他の目的は、ユーザが部分的または完全に覆われたウィンドウを見つけるのを助けるためにコンピュータのディスプレイ・スクリーン上にウィンドウを区別的に表示することにある。

【0008】 この発明の他の目的は、ユーザが部分的または完全に覆われたウィンドウを見つけるのを助けるためにコンピュータのディスプレイ・スクリーン上にウィンドウを比例的に表示することにある。

【0009】 上記及び他の目的は、ここに開示される、コンピュータのディスプレイ・スクリーン上にウィンドウを比例的に表示するための方法及び装置によって達成される。

【0010】

【課題を解決するための手段】コンピュータのディスプレイ・スクリーン上にウィンドウを区別的及び比例的に表示するための方法及び装置が開示される。ディスプレイ・スクリーンに表示されるウィンドウのそれぞれが活動状態（「イン・フォーカス状態」としても知られている）にある時間の長さがモニタされる。ユーザからのコマンドを受け取った時、より長い時間活動状態にあったウィンドウは、より短い時間活動状態にあったウィンドウよりもより区別的に表示される。すなわち、ウィンドウは、ウィンドウのそれぞれが活動状態にあった時間の長さに比例するウィンドウ・サイズを有してディスプレイ・スクリーン上に表示される。例えば、40%の時間活動状態にあったウィンドウは、指定されたウィンドウ・タイル表示領域の40%のサイズを有するであろう。最小ウィンドウ・タイル表示しきい値を越えるに十分に長い時間活動状態になかったウィンドウは、指定されたウィンドウ・タイル表示領域の外側または下側にアイコンとして表示され、ウィンドウの活動のパーセントの計算には含まれない。

【0011】ウィンドウのリストは活動の降順1-Nに分類される。ここで、N=ウィンドウの数、である。ウィンドウ・タイル表示領域はまず、二つの領域、すなわちH1領域とH2領域とに分割される。もしウィンドウ・タイル表示領域の幅がその高さ以上であれば、この分割は垂直方向に行われ、そうでなければ水平方向に行われる。この分割は、第1から第Nのウィンドウが活動状態にあった時間の長さに対する第1及び第2のウィンドウが活動状態にあった時間の長さにH1領域が比例するように行われ、それによって第1から第Nのウィンドウが活動状態にあった時間の長さに対する第3から第Nのウィンドウが活動状態にあった時間の長さにH2領域が比例するようにする。

【0012】この分割が行われた後、H1領域はQ1領域とQ2領域とに分割される。もしH1領域の幅がその高さ以上であれば、この分割は垂直方向に行われ、そうでなければ水平方向に行われる。この分割は、第1及び第2のウィンドウが活動状態にあった時間の長さに対する第1のウィンドウが活動状態にあった時間の長さにQ1領域が比例するように行われる。

【0013】この分割が行われた後、H2領域はQ3領域とQ4領域とに分割される。以前と同様に、もしH2領域の幅がその高さ以上であれば、この分割は垂直方向に行われ、そうでなければ水平方向に行われる。この分割は、第3から第Nのウィンドウが活動状態にあった時間の長さに対する第3のウィンドウが活動状態にあった時間の長さにQ3領域が比例するように行われる。

【0014】第1、第2及び第3の最も高い活動状態にあるウィンドウは、それぞれQ1、Q2及びQ3領域に表示される。もし四つのウィンドウしかなければ、第4のウィンドウはQ4領域に表示される。もし四つよりも

多くのウィンドウがあれば、Q4領域は、必要に応じて、ウィンドウ・タイル表示領域、H1領域及びH2領域が分割されたのと同じ方法で再帰的に分割される。この分割は、最小ウィンドウ・タイル表示しきい値を越えるに十分に長い時間活動状態にあったそれぞれのウィンドウを表示するための領域ができるまで再帰的に継続される。

【0015】

【実施例】図1はこの発明のコンピュータ・システムのブロック図を示す。コンピュータ・システム10は、それぞれシステム・ユニット11に接続されたディスプレイ17、キーボード18及び入力装置19を有する。システム・ユニット11は、メモリ13、記憶装置14及びディスプレイ・アダプタ15に接続された処理装置12を含む。処理装置12は、図16～図28のフロー・チャートにおいてより詳細に述べられるように、この発明を実施するために適当にプログラムされている。記憶装置14及びメモリ13は、制御データ30及びウィンドウ・データ40を含む。

【0016】好ましい実施例において、コンピュータ・システム10はIBM社のPS/2であり、ここでは処理装置12はインテル社の80386マイクロプロセッサである。ディスプレイ・アダプタ15はIBM社の8513ディスプレイ・アダプタであり、ディスプレイ17はIBM社の8513ディスプレイである。入力装置19は好ましくはIBM社のマウスであるが、トラック・ボール、ライト・ペンまたは他の入力装置であってもよい。記憶装置14はオペレーティング・システム・ソフトウェアを含み、これはWordPerfect for Presentation Managerのような好ましくは一つまたはそれ以上のOS/2アプリケーション・プログラムまたはマイクロソフト社のWord for WindowsのようなオプションのDOSアプリケーション・プログラムはもちろん、好ましくはPresentation Managerを有するがマイクロソフト社のWindows 3.0はオプションであるOS/2であってもよい。実行時には、これらのプログラムは部分的または完全にメモリ13内に導入され、処理装置12により実行される。

【0017】コンピュータ・システム10は、アップル・コンピュータ社のマッキントッシュのような他のマイクロコンピュータ、IBM社のAS/400のようなミニコンピュータまたはIBM社の390のようなメイン・フレーム・コンピュータなどの他のタイプのコンピュータ・システムであってもよく、いずれもこの発明の思想及び範囲に含まれる。さらに、コンピュータ・システム10は、IBM社のAS/400のようなより大型のコンピュータ・システムに接続された上述のようなマイクロコンピュータとすることもできる。

【0018】ディスプレイ17はウィンドウ21～26を含む。この発明の目的のために、「ウィンドウ」また

は表示窓はディスプレイ・スクリーンのほぼ全体からディスプレイ・スクリーンの非常に小さな部分までのどこをも占めることができ、OS/2のようなマルチタスキング環境やDOSのような単一タスキング環境における他のウィンドウとともに表示することができる。ウィンドウの数が増加するにつれて、図1のディスプレイ17に示されているように、多くのウィンドウが他のウィンドウによって部分的または完全に覆われるようになる。

【0019】図2は図1のディスプレイ17上のウィンドウ21～26をより詳細に示す。図2は、ユーザがそのコンピュータ上で数分間または数時間タスクを実行した後に典型的なディスプレイ・スクリーンがどのように見えるかの例である。すなわち、我々のユーザが、年間納税計画のある目的を実行するためにそのコンピュータを使用している。6個のウィンドウが示されているが、ディスプレイ・スクリーン上には、1個から数十個のウィンドウを表示することができる。なお、ウィンドウ25は残りのウィンドウよりもより暗い境界線を有し、他のウィンドウのいずれも重なっていない。これは、ウィンドウ25が活動状態のウィンドウであるか、または、「イン・フォーカス状態」にあると考えられるかをユーザに示す。ウィンドウが「イン・フォーカス状態」にある時には、ユーザは、そのウィンドウに含まれているデータを入力し、または操作することができる。

【0020】ウィンドウ21、22、23、24及び26は、他のウィンドウによって少なくとも部分的に覆われている。ウィンドウ22は完全に覆われている。ウィンドウ24及び26はほとんど完全に覆われているが、もしこれらのウィンドウに含まれているデータのいずれかがユーザに表示されるならばほとんど覆われない。

【0021】我々の架空のユーザである納税者タミー(Tammy)は、クリスマスの直前の土曜日の早朝に(彼女は8月にクリスマスの買物を済ませていた)、年間納税計画の彼女の目的の実行を開始した。タミーは、彼女の納税計画を手助けするために数個のアプリケーション・プログラムを同時に使用する。タミーはExcel及びLotus上にスプレッドシート・データを有し、WordPerfect上でIRS宛の手紙と彼女の会計係宛のメモとを書いており、Quicken上に彼女の財務情報を有し、TurboTax上に彼女の税金データを入力する。彼女はまた、File Manager(ファイル・マネージャ)として知られているOS/2アプリケーション・プログラムを使用しており、これはタミーがOS/2上でディレクトリ及びファイルの他の特徴を管理するのを助ける。

【0022】タミーはウィンドウからウィンドウへとぐりぐりとジャンプを行い、彼女がPresentation Managerに関して特に好きな特徴—ウィンドウ25のデータの操作に最も最近時間を費やした(すなわち、Lotus 1-2-3上でスプレッドシートを使用して)。しかし今、タミーは朝の多くの時間を使用したプログラム、すなわちQu

ickenに戻りたい。タミーはディスプレイを手早くスキャンするが、Quickenを含むウィンドウがどこにあるかは簡単にはわからないことを発見するだけである。彼女は、ウィンドウ21、23及び25を、これらのウィンドウがQuickenを含んでいないことを知るために、十分に見ることができる。しかし、Quickenは、部分的に覆われたウィンドウ24または26や完全に覆われたウィンドウ22の中にあることがある。タミーは試行錯誤でそれぞれのウィンドウの中を見るが(覆われたウィンドウ上にマウス・ポインタを移動し、マウスのボタンをダブル・クリックしてウィンドウを活動状態にする)、この技術は非常に煩わしく、スクリーン上に多数のウィンドウがある時や一つ以上のウィンドウが完全に覆われている時には使用されない。

【0023】タミーにとって幸運なことに、彼女のコンピュータ・システムはこの発明のコンピュータ・システム10である。このため、彼女は、彼女のウィンドウを、それらが活動状態または「イン・フォーカス状態」にあった時間の長さに比例するサイズに表示されるように迅速にサイズ変更する能力を有する。後により詳細に述べられるように、図16～図28のフロー・チャートを実行するように適当にプログラムされた、コンピュータ・システム10の処理装置12は、ディスプレイ・スクリーンに表示されたウィンドウのそれぞれが活動状態または「イン・フォーカス状態」にある時間の長さをモニタする。タミーがディスプレイ・ウィンドウ・タイル表示モードを選択した時には、より長い時間活動状態にあったウィンドウは、より短い時間活動状態にあったウィンドウよりもより区別的に表示される。活動状態のウィンドウがより区別的に表示される実際の方法は、ディスプレイ・ウィンドウ・タイル表示モードに対してタミーが選択するパラメータに依存する。

【0024】もしタミーがディスプレイ・ウィンドウ・タイル表示モードの比例した最適化パラメータを得ることができれば、彼女はこの発明の好ましい実施例を使用しており、ディスプレイ・スクリーン17は図8に示すように見える。コンピュータ・システム10は、彼女が今日コンピュータの使用を開始して以来タミーの活動をモニタしてきており、彼女が40%の時間をTurbotaxに、35%の時間をQuickenに、10%の時間をExcelに、7%の時間をWordPerfectに、5%の時間をLotus 1-2-3に、3%の時間をFile Managerに使用したことを知っている。コンピュータ・システム10は、図4～図7に示されているように、指定されたウィンドウ・タイル表示領域を、ウィンドウのそれぞれが活動状態にあった時間の長さに比例する複数の領域に分割する。最適化パラメータが得られたので、それぞれのセグメントは、もしそのセグメントの幅がその高さ以上であれば垂直方向に分割され、そうでなければ水平方向に分割される。File Managerは、5%の指定された最小しきい値ウ

インドウ・サイズを越えなかったので、ウィンドウとして示されていない。従って、File Managerは、ウィンドウ・タイル表示領域の外側にアイコンとして表示される。他のウィンドウの比例する領域のサイズを決定する時には、File Managerが活動状態にあった時間は無視される。

【0025】もしタミーが比例するパラメータを得ることができたが、最適化パラメータは得ることができなかったならば、彼女はこの発明の第1の他の実施例を使用しており、ディスプレイ・スクリーン17は図9に示されるように見える。以前のように、コンピュータ・システム10は、指定されたウィンドウ・タイル表示領域を、ウィンドウのそれぞれが活動状態にあった時間の長さに比例する複数の領域に分割する。最適化パラメータが得られなかったので、ウィンドウ・タイル表示領域が必要とされる数の領域に分割されるまで、第1のセグメントが垂直方向に分割され、その次の「子」領域が水平方向に分割され、次に垂直方向に分割される、等々される。以前のように、File Managerはウィンドウ・タイル表示領域の外側にアイコンとして表示される。

【0026】もしタミーが比例するパラメータを得ることができなかったが、最適化パラメータは得ることができたならば、彼女はこの発明の第2の他の実施例を使用しており、ディスプレイ・スクリーン17は図10に示されるように見える。以前と違って、コンピュータ・システム10は、指定されたウィンドウ・タイル表示領域を四つの等しい領域に分割する。最適化パラメータが得られたので、それぞれのセグメントは、もしそのセグメントの幅がその高さ以上であれば、垂直方向に分割され、そうでないならば水平方向に分割される。以前のように、File Managerはウィンドウ・タイル表示領域の外側にアイコンとして表示される。

【0027】もしタミーが比例するパラメータを得ることができず、最適化パラメータも得ることができなかったならば、彼女はこの発明の第3の他の実施例を使用しており、ディスプレイ・スクリーン17は図11に示されるように見える。コンピュータ・システム10は、指定されたタイル表示領域を四つの等しい領域に分割する。最適化パラメータが得られなかったので、ウィンドウ・タイル表示領域が必要とされる数の領域に分割されるまで、第1のセグメントは垂直方向に分割され、第2のセグメントは水平方向に分割される、等々される。以前のように、File Managerはウィンドウ・タイル表示領域の外側にアイコンとして表示される。

【0028】英語圏の文化では、図8～図11に示されているように左から右にウィンドウを構成することが所望の区別性の順序でウィンドウを構成する最適な方法であると考えるが、比例するサイズであろうとなかろうと、他の文化では右から左へのアプローチを好むかも知れない。後にわかるように、この発明は、これらの文化

の違いに適合するために多少変更することができる。

【0029】図12は図1の制御データ30をより詳細に示す。好ましい実施例においては、後述のように、制御データ30は記憶装置14に記憶され、メモリ13に読み取られる。制御データ30は、この発明のウィンドウ・タイミング機能を実行するために、図16～図28のフロー・チャートによって使用され、更新される情報を含む。

【0030】オン／オフ・フラグ31は、この発明のウィンドウ・タイミング機能がオンであるかオフであるかの情報を絶えず得る。タイマ32は現在のシステム・タイマの値の情報を絶えず得る。好ましい実施例においては、タイマ32は、タイマがスタートまたはリセットされてから経過した時間間隔の数（サンプリング・レート33によって決定される）を表す9桁の値である。中断フラグ34は、後により詳細に述べられるように、ウィンドウ・タイミング機能が中断されたか否かの情報を絶えず得る。最後のイベント・フラグ36は、ユーザの活動をモニタするために使用される。このデータは、ウィンドウが長い時間イン・フォーカス状態にあるが、ユーザから生じる活動はない状態（すなわち、コーヒー・ブレイクなど）を検査し、指定された非活動タイム・アウト期間が経過した時にウィンドウ・タイミング機能を自動的に中断するために使用される。非活動タイム・アウト・フラグ37は、指定された非活動タイム・アウト期間を含む。保管フラグ38は、ユーザがウィンドウ・データ40を保存したいか否かの情報を絶えず得る。

【0031】タイル表示領域51は、ウィンドウ・タイル表示操作のために保管されたスクリーンの領域を示す。最初の二つの数はタイル表示領域の上部左の角のx及びy座標を示す。ここで、ディスプレイ・スクリーンの下部左の角は、図3に示されるように(0, 0)である。もちろん、下部左の角以外の他の原点を選択することもできる。最後の二つの数は、タイル表示領域の下部右の角のx及びy座標を示す。好ましい実施例においては、これらの座標は文字の代わりにミリメートルで測られ、従ってディスプレイの高さと幅との間の適切なアスペクト比を保つことができる。画素のような他の測定単位を使用することもできる。好ましい実施例においては、ディスプレイ17は、幅が約200mm、高さが約155mmの表示領域を有するIBM社の8513ディスプレイである。従って、この表示領域の上部左の角のx及びy座標は(0, 155)であり、下部右の角のx及びy座標は(200, 0)である。図3に示されている我々の例のスクリーンにおいては、タイル表示領域62は、座標が(0, 140)の上部左の角63と座標が(200, 0)の下部右の角64とを有するものとして示されている。従って、制御データ30のタイル表示領域51は、0, 140, 200, 0を含む。非タイル表示領域61は、アイコン及び他の非タイル表示の使用の

ために、ディスプレイ・スクリーン上に保管される。

【0032】領域52は比例タイル表示がオンであるかオフであるかの指標を含む。もし領域52がオンであれば、この発明は、ウィンドウが活動状態にあった時間の長さに基づいてタイル表示領域を比例的に分割する。もし領域52がオフであれば、この発明は、タイル表示領域を四つの等しい象限に分割し、次に第4の象限を、最小タイル表示のパーセントを越えるそれぞれのウィンドウに対して一つの領域ができるまで再帰的に分割する。好ましい実施例においては、この領域はオンである。

【0033】領域53は最適化がオンであるかオフであるかの指標を含む。もし領域53がオンであれば、この発明は、もしセグメントの幅がその高さ以上であればセグメントを垂直方向に分割し、もしセグメントの幅がその高さよりも小さければセグメントを水平方向に分割する。もし領域53がオフであれば、この発明は、セグメントの垂直方向の分割と水平方向の分割とを逆にする。好ましい実施例においては、この領域はオンである。

【0034】領域54は最小タイル表示パーセントの指標を含む。最小タイル表示パーセントよりも少ない時間活動状態にあったウィンドウは、ウィンドウの代わりにアイコンとして表示される。これは、ウィンドウが小さくなり過ぎて読み取り不能となったり使用不能となったりしないように行われる。例えば、もしウィンドウが1000の時間間隔のうちの30の時間間隔の間（または3%の時間）活動状態にあったとすると、このウィンドウは、もし5%の最小タイル表示パーセントが領域54で指定されれば、アイコンとして表示される。なお、アイコンは、スクリーン上の他の得られるスペースの中から、もし得られるスペースがなければウィンドウ・タイル表示領域の下側に、ウィンドウ・タイル表示領域の外側に表示される。

【0035】図13は図1のウィンドウ・データ40をより詳細に示す。好ましい実施例においては、後述のように、ウィンドウ・データ40はユーザの選択によって記憶装置14に記憶され、メモリ13に読み取られる。ウィンドウ・データ40は、この発明のウィンドウ・タイミング機能及びウィンドウ・タイル表示機能を実行するために、図16～図28のフロー・チャートによって使用され、更新される情報を含む。ウィンドウ・データ40はカラム41～44に配置される。この発明のウィンドウ・タイミング機能がオンであった時にある時点で活動状態にあったそれぞれのオープン・ウィンドウは、メモリ13内のウィンドウ・データ40に含まれる。カラム41はこれらのウィンドウの名称または他の識別子を含む。カラム42はウィンドウのそれぞれがイン・フォーカス状態とされた最後の時のタイマ32の値を含む。カラム43は、ウィンドウ・データ40の中のウィンドウのそれぞれがイン・フォーカス状態にあった時間間隔の総数を含む。カラム44は、図23～図27のフロー

・チャートによって決定されるように、表示すべきウィンドウの上部左及び下部右の角のx及びy座標を含む。

【0036】図14は、活動の降順に分類された後のウィンドウ・データ40を示す。

【0037】図15はウィンドウ・タイミング機能パラメータを示す。これらのパラメータには通常はデフォルト値が割り当てられるが、可能な変更に対する要求の際にユーザに表示することができる。第1のパラメータは、ウィンドウ・タイミング機能がオンであるかオフであるかについて質問する。ウィンドウがより一般的な方法で動作することをユーザが好む時がある。次のパラメータはサンプリング・レートである。これは、ユーザがウィンドウ・タイミング機能の細分性のレベルを制御することを可能とする。次のパラメータは非活動タイム・アウト期間を指定する。次のパラメータは、ユーザがウィンドウ・タイミングを中断したいかどうかについて質問する。このパラメータは、図15に示されているスクリーンによって選択することができ、あるいは、このパラメータをオンまたはオフに切り替えるために特殊なキー配列とすることもできる。このパラメータは、浴室での休息または他の中断の影響を最小にするのに非常に役に立つ。次のパラメータは、この計算セッションの間に生成されたウィンドウ・データ40を次の計算セッションのために保管すべきか否かについて質問する。もしそうであれば、ウィンドウ・データ40は、ウィンドウがクローズされる時に、メモリ13から記憶装置14に書き込まれる。次のパラメータは、ウィンドウ・タイミングをリセットすべきか否かについて質問する。特にもしユーザが現在、以前に実行されたものと完全に無関係のタスクを実行しているならば、計算セッションの中間で開始するのが望ましい。もしユーザがウィンドウ・タイミングをリセットすべきであることを指定すれば、全てのウィンドウは、あたかもそれらがこのセッションの間活動状態になかったかのように新たに開始する。

【0038】次のパラメータは、所望のタイル表示領域の上部左及び下部右の角のx及びy座標をユーザに促す。次のパラメータは、比例タイル表示がオンであるべきかオフであるべきかについて質問する。次のパラメータは、最適化がオンであるべきかオフであるべきかについて質問する。最後のパラメータは、ユーザが最小タイル表示パーセントを指定することを可能とする。

【0039】次に、図16～図28のフロー・チャートに示されているこの発明の動作をより詳細に説明する。図16～図20を参照すると、ブロック101は制御データ30を記憶装置14からメモリ13にロードする。ブロック102はタイマ32と制御データ30における最後のイベント・フラグ36とを初期化する。ブロック103はタイマ32をスタートさせる。これは、図22のフロー・チャートの実行を初期化することによって行われる。図22を参照すると、ブロック201は、図1

6のブロック198からタイマを停止するための何らかの指標を受け取ったか否かについて質問する。もし受け取ったならば、プログラムはブロック299で終了する。もし受け取っていないならば、ブロック202は制御データ30におけるサンプリング・レート33が経過するのを待つ。ブロック205は、制御データ30における中断フラグ34がFALSEであるか否かを見るために検査を行う。もしこのフラグがFALSEでなければ(好ましい実施例においてはTRUEまたはTRUE2)、これはウィンドウ・タイミングを中断すべきであることの指標である。後により詳細に述べられるように、もし彼女がサンプリングを中断したいことをユーザが示したならば、あるいは、非活動タイム・アウト期間が終了したならば、この条件は存在しうる。もしこの条件が存在しないならば、制御のフローはループをブロック201に戻る。もしブロック205が中断がFALSEであることを示せば、ブロック208は、一つ以上のタイマ期間が経過したことを示すために、制御データ30におけるタイマ32を1だけ増加させる。

【0040】再び図16を参照すると、ブロック103がタイマをスタートさせた後、ブロック110は処理すべきウィンドウ・イベントがあるか否かを見るために検査を行う。好ましい実施例においては、「ウィンドウ・イベント」は、この発明によって生成されるイベントとともに、ウィンドウにデータを入力したり、テキスト・カーソルまたはマウス・カーソルを移動したり、スクロール・バー上にクリックしたりするなどのような、Presentation Managerによって生成された任意のイベントである。もしブロック110が否定的に答えられるならば、ブロック112は全てのウィンドウがクローズされたか否かを見るために検査を行う。もしそうであれば、ブロック198はタイマ32を停止し、もし保管フラグ38がオンであるならば、記憶装置14に制御データ30を書き込む。プログラムはその後、ブロック199で終了する。もしブロック112が全てのウィンドウがクローズされなかったことを示すならば、ブロック115は、制御データ30における非活動タイム・アウト37が経過したか否かを見るために検査を行う。これは、タイマ32から最後のイベント36を減じ、その結果にサンプリング・レート33を乗じ、60で除することによって行われる。もしこの結果が非活動タイム・アウト期間37における値よりも大きければ、ブロック115は肯定的に答えられ、ブロック118は中断ウィンドウ・イベントを自動的に生成する。いずれにしても、制御のフローはブロック110に戻る。

【0041】処理すべきウィンドウ・イベントがあることをブロック110が示す時、ブロック104はこれがオープン・ウィンドウ・イベントであるか否かを見るために検査を行う。もしそうであれば、ブロック105は、同一の名前を持つウィンドウがメモリ13内にすで

にあるか否かを見るために検査を行う。もしそうであれば、このウィンドウはブロック106において新たな名前(すなわち、TurboTax2)を与えられる。いずれにしても、もし何らかのそのようなデータが前のセッションから保管され、及び、もし保管フラグ38がオンであれば、ブロック107は記憶装置14からのレコードにおけるこのウィンドウをメモリ13にロードする。その後、ブロック108において通常のウィンドウ処理が実行される。

【0042】ブロック121はこれがクローズ・ウィンドウ・イベントであったか否かを見るために検査を行う。もしそうであれば、メモリ13におけるウィンドウ・データ40からウィンドウを、ユーザがウィンドウ・ディスプレイ・モードを選択した時にそれが再び現れないようにするために取り除くのが適当である。この機能はブロック122によって実行される。ブロック122はまた、もし保管フラグ38がオンであれば、このウィンドウ・レコードを記憶装置14に書き込む。ブロック123は次に、このイベントに対する通常のウィンドウ処理を実行する。

【0043】もしブロック121が否定的に答えられれば、ブロック120はこのイベントがゲット・フォーカス・イベントであるか否かについて質問する。好ましい実施例においては、「ゲット・フォーカス」イベントは、ウィンドウが活動状態にされ、あるいは「イン・フォーカス状態」にされる時にはいつでも、Presentation Managerによって生成される。もしブロック120が肯定的に答えられれば、ブロック125はフォーカスが合うウィンドウを登録する。この登録は、図21のフロー・チャートを活性化させることによって行われる。

【0044】図21を参照すると、ブロック301はウィンドウ・データ30におけるオン/オフ・フラグ31がオンであるか否かを見るために検査を行う。もしオンでなければ、プログラムはブロック399で直ちに終了する。もしこのフラグがオンであれば、ブロック303は現在の時刻を制御データ30におけるタイマ32から得る。ブロック304は、イン・フォーカス状態にすべきウィンドウがウィンドウ・データ40に存在するか否かを見るために検査を行う。もし存在しなければ、ブロック306はウィンドウ・データ40におけるこのウィンドウに対する新たなレコードを生成する。イン・フォーカス・カラム42、全カラム43及び領域カラム44にはゼロが配置される。もしブロック304が否定的に答えられれば、ブロック308は、ウィンドウ・データ40においてイン・フォーカス状態とすべきこのウィンドウに関連するウィンドウ・レコードを使用する。

【0045】ブロック310は、これが「ゲット・フォーカス」ウィンドウ・イベントであるか「ルーズ・フォーカス」ウィンドウ・イベントであるかについて質問する。我々のイベントは「ゲット・フォーカス」イベント

であるから、ブロック315は制御データ30からのタイム32の現在の値をこのウィンドウに対するイン・フォーカス・カラム42に入れ、プログラムはブロック399で終了する。

【0046】図16及び図17を参照すると、ブロック125が図21のフロー・チャートを活性化することによってフォーカスが合うウィンドウを登録し、ブロック126はこのイベントに対する通常のウィンドウ処理を実行する。好ましい実施例においては、Presentation Managerが選択されたウィンドウをイン・フォーカス状態にする。

【0047】もしブロック120が否定的に答えられれば、ブロック130はこれがルーズ・フォーカス・ウィンドウ・イベントであるか否かについて質問する。好ましい実施例においては、「ルーズ・フォーカス」イベントは、他のウィンドウがイン・フォーカス状態とされているためにウィンドウがもはや活動状態にない時にはいつでも、Presentation Managerによって生成される。もしブロック130が肯定的に答えられれば、ブロック135はフォーカスを失ったウィンドウを登録する。すでに述べたように、この登録は、ブロック310（図21）がこれがルーズ・フォーカス・イベントであることを決定し、ブロック315の代わりにブロック320が実行されることを除いて、図21のフロー・チャートを再び活性化することによって行われる。ブロック320は、どれ位の間それが活動状態にあったかを示すために、このウィンドウに対するウィンドウ・データ40の全カラム43における値を更新する。このウィンドウに対するイン・フォーカス・カラム42に含まれる値が、制御データ30におけるタイム32の現在の値から減じられる。この結果は、このウィンドウに対する全カラム43における現在の値に加えられ、その和がこのウィンドウに対する全カラム43に置かれる。

【0048】再び図16及び図17を参照すると、ブロック135が図21のフロー・チャートを活性化することによってフォーカスを失ったウィンドウを登録した後、ブロック136はこのイベントに対する通常のウィンドウ処理を実行する。好ましい実施例においては、Presentation Managerは選択を解除されたウィンドウからフォーカスを外す。

【0049】もしブロック130が否定的に答えられれば、ブロック150（図18及び図19）は中断ウィンドウ・イベントが生成されたか否かを見るために検査を行う。図16及び図17のブロック118（タイム・アウト期間が終了）によって、あるいはもしユーザが図15における彼女のメニューにおいてタイミングを中断させなければならないことを示したならば、このイベントは生成することができる。もしイベントがユーザによって生成されたならば、ブロック151は制御データ30における中断フラグ34をTRUEに設定する。もしイ

ベントがブロック118によって生成されたならば、ブロック151は制御データ30における中断フラグ34をTRUEに設定する。いずれにしても、この結果、図22のフロー・チャートを独立に実行するブロック205は否定的に答えられ、それによってブロック208が飛ばされる。

【0050】再び図18及び図19を参照すると、もしブロック150が否定的に答えられれば、ブロック155は再開ウィンドウ・イベントが生成されたか否かを見るために検査を行う。図28のブロック1050（タイム・アウト期間は終了したが、ユーザは今ユーザの活動によって生じたウィンドウ・イベントを実行した）によって、あるいはもしユーザが図15における彼女のメニューにおいてタイミングを中断しなければならないことを示したならば、このイベントは生成することができる。いずれにしても、ブロック156は制御データ30における中断フラグ34をFALSEに設定する。この結果、図22のフロー・チャートを独立に実行するブロック205が肯定的に答えられ、それによってブロック208が実行される。

【0051】再び図18及び図19を参照すると、もしブロック155が否定的に答えられれば、ブロック160はリセット・ウィンドウ・イベントが生成されたか否かを見るために検査を行う。もしユーザが図15における彼女のメニューにおいてタイミングをリセットしなければならないことを示したならば、このイベントは生成される。ブロック161はウィンドウ・データ40における全てのウィンドウを通してループをつくり、ブロック162はイン・フォーカス・カラム42、全カラム43及び領域カラム44の全ての値をゼロに設定する。処理すべきウィンドウ・データ40にウィンドウ・レコードがもうない時には、ブロック161は否定的に答えられ、制御のフローはブロック163に移る。

【0052】ブロック163は保管ウィンドウ・イベントが生成されたか否かを見るために検査を行う。もし図15における彼女のメニューにおいて保管パラメータの値を変更したならば、このイベントは生成される。もし変更が行われれば、ブロック164は制御データ30における保管フラグ38を、ユーザによって指定されるようにオンまたはオフに設定する。

【0053】ブロック165は、ウィンドウ・タイミング・ターン・オフ・イベントが生成されたか否かを見るために検査を行う。もしユーザが図15における彼女のメニューにおいてウィンドウ・タイミングを停止しなければならないことを示したならば、このイベントは生成される。もしそうであれば、ブロック166は制御データ30におけるオン／オフ・フラグ31をオフに設定する。ブロック168は、ウィンドウ・データ40における全てのウィンドウを通してループをつくり、ブロック169はイン・フォーカス・カラム42及び全カラム4

3の全ての値をゼロに設定する。処理すべきウィンドウ・データ40にウィンドウ・レコードがもうない時には、ブロック168は否定的に答えられ、制御のフローはブロック170に移る。

【0054】ブロック170は、ウィンドウ・タイミング・ターン・オン・イベントが生成されたか否かを見るために検査を行う。もしユーザが図15における彼女のメニューにおいてウィンドウ・タイミングをオンしなければならないことを示したならば、このイベントは生成される。もしそうであれば、ブロック171は制御データ30におけるオン／オフ・フラグ31をオンに設定する。いずれにしても、制御のフローはブロック175に移る。

【0055】ブロック175は、サンプリング・レート設定ウィンドウ・イベントが生成されたか否かを見るために検査を行う。もしユーザが図15における彼女のメニューにおいてサンプリング・レートに対する値をうめたならば、このイベントは生成される。もしそうであれば、ブロック176は制御データ30におけるサンプリング・レート領域33をユーザによって設定された値に設定する。

【0056】ブロック177は、タイル表示領域定義ウィンドウ・イベントが生成されたか否かを見るために検査を行う。もしユーザが図15における彼女のメニューにおいてタイル表示領域の上部左及び下部右の角のx及びy座標をうめたならば、このイベントは生成される。もしそうであれば、ブロック178は制御データ30の領域51に対するタイル表示領域の座標をユーザによって設定された値に入力する。

【0057】ブロック180は、比例タイル表示領域イネーブル・イベントが生成されたか否かを見るために検査を行う。もしユーザが図15における彼女のメニューにおいてウィンドウ・タイル表示モードの比例タイル表示パラメータをオンしなければならないことを示したならば、このイベントは生成される。もしそうであれば、ブロック181は制御データ30における比例タイル表示フラグ52をオンに設定する。

【0058】ブロック182は、比例タイル表示ディスエーブル・ウィンドウ・イベントが生成されたか否かを見るために検査を行う。もしユーザが図15における彼女のメニューにおいてウィンドウ・タイル表示モードの比例タイル表示パラメータをオフしなければならないことを示したならば、このイベントは生成される。もしそうであれば、ブロック183は制御データ30における比例タイル表示フラグ52をオフに設定する。

【0059】ブロック185は、最適化イネーブル・ウィンドウ・イベントが生成されたか否かを見るために検査を行う。もしユーザが図15における彼女のメニューにおいてウィンドウ・タイル表示モードの最適化パラメータをオンしなければならないことを示したならば、こ

のイベントは生成される。もしそうであれば、ブロック186は制御データ30における最適化フラグ53をオンに設定する。

【0060】ブロック187は、最適化ディスエーブル・ウィンドウ・イベントが生成されたか否かを見るために検査を行う。もしユーザが図15における彼女のメニューにおいてウィンドウ・タイル表示モードの最適化パラメータをオフしなければならないことを示したならば、このイベントは生成される。もしそうであれば、ブロック183は制御データ30における最適化フラグ53をオフに設定する。

【0061】ブロック190は、最小タイル表示パーセント設定ウィンドウ・イベントが生成されたか否かを見るために検査を行う。もしユーザが図15における彼女のメニューにおいて最小タイル表示パーセントに対する値をうめたならば、このイベントは生成される。もしそうであれば、ブロック191は制御データ30における最小タイル表示パーセント領域54をユーザによって設定された値に設定する。

【0062】ブロック195は、処理すべき他のウィンドウ・イベントがあるか否かを見るために検査を行う。もしそうであれば、ブロック196はこのイベントに対する通常のウィンドウ処理を実行する。いずれにしても、制御のフローはループを図16及び図17のブロック115に戻る。

【0063】図16～図19及び図22のフロー・チャートは独立に実行しているが、図23のフロー・チャートも処理装置12内で独立に実行している。このフロー・チャートは、図8～図11に示されているように、ユーザがこの発明のウィンドウ・タイミング機能を使用して彼女のウィンドウを再配置したいか否かを見るためにユーザ入力をモニタする。図23を参照すると、ブロック401は、ユーザがディスプレイ・ウィンドウ・タイル表示モードを選択したか否かを見るために検査を行う。もしブロック401がディスプレイ・ウィンドウ・タイル表示モードが選択されなかったと決定したならば、ブロック403は全てのウィンドウがクローズされたか否かを見るために検査を行う。もしそうであれば、プログラムはブロック499で終了する。もしそうでなければ、プログラムは、ディスプレイ・ウィンドウ・タイル表示モードが選択されたか否かを見るための検査を再び行うためにループをブロック401に戻る。好ましい実施例においては、ディスプレイ・ウィンドウ・タイル表示モードは指定されたキー配列によって選択される。例えば、ALT-Wを使用することができる。あるいは、マウス・ボタンの組み合わせを使用することもでき、ユーザはメニューや、ディスプレイ・スクリーン上のボタンのアイコンまたは表示上でクリックを行うことによってモードを選択することもできる。いずれにしても、ブロック401がディスプレイ・ウィンドウ・タイ

ル表示モードが選択されたことを決定すると直ちに、ブロック407は、現在イン・フォーカス状態にあるウィンドウに対する全カラム43を、タイマ32の現在の値からイン・フォーカス状態にあるカラム42の値を減じたものに等しく設定し、この結果を全カラム43に現在ある値に加える。この機能は、図21のブロック320によって実行されるものと同一であり、現在イン・フォーカス状態にあるウィンドウに関する最も最新の情報が使用されることを保証する。ブロック407はまた、このウィンドウに対するイン・フォーカス状態のカラム42をタイマ32の現在の値に等しく設定する。

【0064】ブロック901は、通常の方法で全てのディスプレイの表示を消去する。ブロック903は、ウィンドウ・データ40におけるウィンドウ・レコードを、全カラム43における値によって、最も高いものから最も低いものに、活動の順に分類する。従って、最も高い活動状態にあるウィンドウはウィンドウ・データ40の最上部における第1のレコードとなり、最も低い活動状態にあるウィンドウはウィンドウ・データ40の底部における最後のレコードとなるであろう。ブロック905は、いずれかのウィンドウが最小しきい値パーセントよりも活動状態が低かったか否かを見るために、ウィンドウ・データ40におけるウィンドウ・レコードの検査を行う。これは、全ての活動状態にあるウィンドウの非識別制御データ30がそれぞれのウィンドウが活動状態にあったパーセントに達するための全時間でそれぞれのウィンドウに対するウィンドウ・データ40における全カラム43の値を除し、これに100を乗じ、この値を制御データ30の領域54に含まれている最小しきい値パーセントと比較することによって、行われる。ブロック905がこの最小しきい値パーセント以下であると決定するそれぞれのウィンドウに対して、ブロック905は、後の使用のために制御データ40の領域カラム44に「ICON」を入れる。制御のフローは次にブロック1100に移り、ここで図24のウィンドウ・タイル表示領域計算サブルーチンが呼び出される。活動によって分類された、処理すべき全てのウィンドウのウィンドウ・リストはこのサブルーチンに渡される。

【0065】図24のウィンドウ・タイル表示計算サブルーチンは計算を行い、領域カラム44における値「ICON」をまだ有していないそれぞれのウィンドウに対するウィンドウ・データ40の領域カラムに対する値をうめる。ブロック1101は四つの象限の重みを割り当てることによって開始する。第1の象限(Q1wt)は、ウィンドウ・データ40における第1のウィンドウ・レコード(すなわち、最も高い活動状態にある)に対する全カラム43に含まれている値を与えられる。同様に、第2の象限(Q2wt)は、ウィンドウ・データ40における第2のウィンドウ・レコード(すなわち、次に高い活動状態にある)に対する全カラム43に含まれ

ている値を与えられる。第3の象限(Q3wt)は、ウィンドウ・データ40における第3のウィンドウ・レコード(すなわち、最も高い活動状態にある)に対する全カラム43に含まれている値を与えられる。第4の象限(Q4wt)は、ウィンドウ・データ40におけるウィンドウの残りの和に対する全カラム43に含まれている値を与えられる。なお、ブロック1101は、もし「ICON」の表示がウィンドウに対する領域44にあるならば、いずれの象限の重みに対する全カラム43における値も使用しない。その代わりに、これらの重みに「0」が加えられる。「0」はまた、もしウィンドウが存在しなければ、重みに対して使用される。

【0066】ウィンドウ・データ40に示されている我々の例のデータ(図14に示されているように分類されている)を使用して、ブロック1101は、Q1wt、Q2wt、Q3wt及びQ4wtに対して、それぞれ400、350、100及び120の重みを割り当てる。Q4wtは、WordPerfect 及びLotus 1-2-3に対する全カラム43における値を加えることによって決定される(70+50)。ブロック905はこのウィンドウが最小しきい値パーセントを越えなかったことを決定し、カラム44に「ICON」を入れたので、File Managerに対する全カラム43における値(30)は0で置換され、Q4wtに加えられる。

【0067】ブロック1105は次に、H1wtを決定するためにQ1wt及びQ2wtの重みを結合するとともに、H2wtを決定するためにQ3wt及びQ4wtの重みを結合する。我々の例においては、 $H1wt = 750$ ($400 + 350$)、 $H2wt = 220$ ($100 + 120$)である。次に、図25～図27のセグメント分割サブルーチン1200が、ブロック1200Aにおいて初めて呼び出される。以下の入力パラメータがサブルーチンに渡される。すなわち、タイル表示領域の座標(我々の例においては0、140、200、0—図3参照)、H1wt(750)及びH2wt(120)である。出力パラメータの領域H1及び領域H2もまた、サブルーチン1200に渡される。サブルーチン1200は、指定されたセグメント(タイル表示領域)を分割し、それがセグメントを(領域H1及び領域H2)に分割する二つの領域の上部左及び下部右の角のx及びy座標を元に戻す。

【0068】図25～図27を参照すると、ブロック1201は、図24から第1の入力パラメータとして渡されるセグメントの幅及び高さを計算する。幅は、セグメントの下部右の角のx座標(我々の例においては200)からセグメントの上部左の角のx座標(我々の例においては0)を減ずることによって計算される。高さは、セグメントの上部左の角のy座標(我々の例においては140)からセグメントの下部右の角のy座標(我々の例においては0)を減ずることによって計算され

る。従って、ブロック 1201 は、我々のセグメントの幅が 200 であり、高さが 140 であることを決定する。ブロック 1205 は、このセグメントが分割される二つの領域の x 及び y 座標を、セグメント自身の x 及び y 座標と同一の値に初期化する。このため、我々の例においては、両セグメントは、0、140 の上部左 (UL) の x 及び y 座標と 200、0 の下部右 (LR) の x 及び y 座標とを有するであろう。

【0069】ブロック 1210 は、 $Wt1 + Wt2$ が入力パラメータ > 0 として渡されたか否かについて質問する。もし渡されていないならば、これらのセグメントに入れるべきウィンドウがないので、セグメントは二つの領域に分割されず、サブルーチンはブロック 1298 において呼び出された所に戻る。もしブロック 1210 が肯定的に答えられたならば、領域に入れるべきウィンドウは少なくとも一つあり、制御のフローはブロック 1215 に続く。

【0070】ブロック 1215 は、制御データ 30 における比例タイル表示フラグ 52 がオンであるか否かを見るために検査を行う。もしオンであれば、ブロック 1220 は、制御データ 30 における最適化フラグ 53 がオンであるか否かを見るために検査を行う。もしオンであれば、好ましい実施例における同様に、制御のフローはブロック 1230 に移る。ブロック 1230 は、セグメントの幅がその高さ以上であるか否かを見るために検査を行う。もしそうであれば、それはより良く見える

(そして、もしセグメントが垂直方向に分割されるならば、「最適化」されたものである)。これはブロック 1235 によって実行される。ブロック 1235 は、領域 1 の下部右 (LR) の x 座標と領域 2 の上部左 (UL) の x 座標とを、分割すべきセグメントの UL x 座標とセグメントの幅に $Wt1 / (Wt1 + Wt2)$ を乗じたものとを加えたものに等しく設定する。この動作は図 4 に示されている。ブロック 1235 は、領域 1 の LR x 座標を 200 (ブロック 1205 において初期化された) から以下のように変更することによって、セグメント

(タイル表示領域) を分割する。すなわち、領域 1 の $LRx = 0 + (200 \times 750 / (750 + 220)) = 155$ である。領域 2 の UL x 座標も 155 に変更される。なお、Presentation Manager の下においてはウィンドウ座標は通常は整数として表現されなければならないので、領域 1 の LR x 及び領域 2 の UL x 座標の値 155 は実際には 154.63918 を丸めたものである。この丸め演算はウィンドウ領域が ± 0.5 mm の丸め誤差で比例することを意味するが、この発明の目的に対して、これは「比例」と同一であると考えられる。ブロック 1235 が実行された後、サブルーチンはブロック 1299 から図 24 のブロック 1200B に戻り、領域 H1 及び領域 H2 に対する座標 (領域 1 及び領域 2 に対する値を使用する) を出力パラメータとして戻す。

【0071】再び図 24 を参照すると、図 25 ~ 図 27 のセグメント分割サブルーチン 1200 は次に、ブロック 1200B において 2 回目の呼び出しを受ける。以下の入力パラメータがサブルーチンに渡される。すなわち、領域 H1 の座標 (我々の例においては 0、140、155、0 - 図 4 参照)、 $Q1wt$ (400) 及び $Q2wt$ (350) である。出力パラメータの領域 Q1 及び領域 Q2 もサブルーチン 1200 に渡される。サブルーチン 1200 は、領域 H1 を領域 Q1 及び領域 Q2 に分割し、これらの二つの領域の上部左及び下部右の角の x 及び y 座標を戻す。

【0072】再び図 25 ~ 図 27 を参照すると、ブロック 1201 は、以前と同様に、図 24 から第 1 の入力パラメータとして渡されたセグメントの幅及び高さを計算する。ブロック 1201 は、領域 H1 の幅が 140 であり、高さが 140 であると決定する。ブロック 1205 は、このセグメントが分割される二つの領域の x 及び y 座標を、セグメント自身の x 及び y 座標 (我々の例においては 0、140、155、0) と同一の値に初期化する。

【0073】制御のフローは、以前のようにブロック 1215、1220 及び 1230 を通り、ブロック 1235 は領域 H1 を垂直方向に分割する。ブロック 1235 は、領域 1 の下部右 (LR) の x 座標と領域 2 の上部左 (UL) の x 座標とを、分割すべきセグメントの UL x 座標とセグメントの幅に $Wt1 / (Wt1 + Wt2)$ を乗じたものとを加えたものに等しく設定する。この動作は図 5 に示されている。ブロック 1235 は、領域 1 の LR x 座標を 155 (ブロック 1205 において初期化された) から以下のように変更することによって、セグメント (領域 H1) を分割する。すなわち、領域 1 の $LRx = 0 + (155 \times 400 / (400 + 350)) = 83$ である。領域 2 の UL x 座標も 83 に変更される。ブロック 1235 が実行された後、サブルーチンはブロック 1299 から図 24 のブロック 1200C に戻り、領域 Q1 及び領域 Q2 に対する座標 (領域 1 及び領域 2 に対する値を使用する) を出力パラメータとして戻す。

【0074】再び図 24 を参照して、セグメント分割サブルーチン 1200 は次に、ブロック 1200C において 3 回目の呼び出しを受ける。以下の入力パラメータがサブルーチンに渡される。すなわち、領域 H2 の座標 (我々の例においては 155、140、200、0 - 図 5 参照)、 $Q3wt$ (100) 及び $Q4wt$ (120) である。出力パラメータの領域 Q3 及び領域 Q4 も、サブルーチン 1200 に渡される。サブルーチン 1200 は、領域 H2 を領域 Q3 及び領域 Q4 に分割し、これらの二つの領域の上部左及び下部右の角の x 及び y 座標を戻す。

【0075】再び図 25 ~ 図 27 を参照すると、ブロック 1201 は、以前と同様に、図 24 から第 1 の入力パ

ラメータとして渡されたセグメントの幅及び高さを計算する。ブロック1201は、領域H1の幅が45であり、高さが140であることを決定する。ブロック1205は、このセグメントが分割される二つの領域のx及びy座標を、セグメント自身のx及びy座標（我々の例においては155、140、200、0）と同一の値に初期化する。

【0076】制御のフローは以前のようにブロック1215及び1220を通るが、領域H2の幅がその高さよりも小さいので、ブロック1230は今度は否定的に答えられる。従って、制御のフローは、領域H2を水平方向に分割するために、ブロック1240に移る。ブロック1240は、領域1の下部右（LR）のy座標と領域2の上部左（UL）のy座標とを、分割すべきセグメントのULy座標から、セグメントの高さに $Wt1 / (Wt1 + Wt2)$ を乗じたものを減じたものに等しく設定する。この動作は図6に示されている。ブロック1240は、領域1のLRy座標を0（ブロック1205において初期化された）から以下のように変更することによって、セグメント（領域H2）を分割する。すなわち、領域1のLRy座標を $140 - (140 \times 100 / (100 + 120)) = 76$ である。領域2のULy座標も76に変更される。ブロック1240が実行された後、サブルーチンはブロック1299から図24のブロック1115に戻り、領域Q3及び領域Q4に対する座標（領域1及び領域2に対する値を使用する）を出力パラメータとして戻す。

【0077】再び図24を参照すると、ブロック1115は、最初の三つのウィンドウに対して計算された座標をウィンドウ・データ40の領域44に書き込む。我々の例においては、最も高い活動状態にあるウィンドウ（Turbotax）に対する領域44は、領域Q1の座標0、140、83、0でうめられる。次に高い活動状態にあるウィンドウ（Quicken）に対する領域44は、領域Q2の座標83、140、155、0でうめられる。次に高い活動状態にあるウィンドウ（Excel）に対する領域44は、領域Q3の座標155、140、200、76でうめられる。これは図14に示されている。

【0078】ブロック1120は、処理する必要があるウィンドウがまだ存在するか否かについて質問する。これは、もし領域がまだ生成されていなかったウィンドウがあるか否かを見るために、サブルーチン1000に渡されたウィンドウ・リストの検査を行うことによって行われる。我々の例においては我々は処理が残されたウィンドウ・リストにおける第4及び第5のウィンドウを有するので、ブロック1120は肯定的に答えられる。

【0079】ブロック1130は、ブロック1101、1105、1200A、1200B、1200C、1115及び1120を実行することによって、ウィンドウ・タイル表示計算サブルーチン1100を再帰的に繰り

返す。これは、領域44に「ICON」を有しないウィンドウ・データ40におけるそれぞれのウィンドウに対して一つの領域ができるように領域Q4を十分な領域にさらに分割するために行われる。サブルーチン1100が呼び出されるたびに、サブルーチン1100に渡されたウィンドウ・リストは、領域がまだ生成されていなかったウィンドウだけを含む。我々の例においては、サブルーチン1100に渡されたウィンドウ・リストは、図14に示されるように分類された、ウィンドウ・データ40における第4及び第5のウィンドウ（すなわち、Word Perfect MY、TXT及びLotus 1-2-3）を含むであろう。もしサブルーチン1100に渡されたウィンドウ・リストに三つ以上のウィンドウが残されていれば、ブロック1130は、ウィンドウ・リストに三つまたはそれよりも少数のウィンドウが残されるまでサブルーチン1100を多数回呼び出し、その時点でサブルーチン1100は最後の実行を行う。

【0080】我々の例において我々はウィンドウ・リストにサブルーチン1100に渡された二つのウィンドウだけを有しているので、ブロック1130は、サブルーチン1100を1回呼び出す。従って、ブロック1101は再び実行され、象限の重みは以下のように割り当てられる。すなわち、 $Q1wt = 70$ 、 $Q2wt = 50$ 、 $Q3wt = 0$ 及び $Q4wt = 0$ である。ブロック1105は、 $H1wt = 120$ 及び $H2wt = 0$ となるように象限の重みを結合する。セグメント分割サブルーチン1200は次に、ブロック1200Aにおいて再び呼び出される。以下の入力パラメータがサブルーチンに渡される。すなわち、領域Q4の座標（我々の例においては155、76、200、0—図6参照）、 $H1wt (120)$ 及び $H2wt (0)$ である。出力パラメータの領域Q4-H1及び領域Q4-H2もサブルーチン1200に渡される。サブルーチン1200は領域Q4を領域Q4-H1及び領域Q4-H2に分割し、これらの二つの領域の上部左及び下部右の角のx及びy座標を戻す。

【0081】再び図25～図27を参照すると、ブロック1201は、以前のように、図24から第1の入力パラメータとして渡されたセグメントの幅及び高さを計算する。ブロック1201は、領域Q4の幅が45であり、高さが76であると決定する。ブロック1205は、このセグメントが分割される二つの領域のx及びy座標を、セグメント自身のx及びy座標（我々の例においては155、76、200、0）と同一の値に初期化する。

【0082】制御のフローは以前のようにブロック1215及び1220を通るが、領域Q4の幅がその高さよりも小さいので、ブロック1230は今度は否定的に答えられる。従って、制御のフローは、領域Q4を水平方向に分割するために、ブロック1240に移る。ブロック1240は、領域1の下部右（LR）のy座標と領域

2の上部左(UL)のy座標とを、分割すべきセグメントのULy座標から、セグメントの高さに $Wt1 / (Wt1 + Wt2)$ を乗じたものを減じたものに等しく設定する。この動作は図6に示されている。ブロック1240は、領域1のLRy座標を0(ブロック1205において初期化された)から以下のように変更することによって、セグメント(領域Q4)を分割することを試みる。すなわち、領域1のLRy $=76 - (76 \times 120 / (120 + 0)) = 0$ である。領域2のULy座標も同様に0に変更される。なお、ULy座標及びLRy座標は両方とも0であるので、領域2は全く幅を有していない。これは、この領域に入ってくるウィンドウはなく、この領域はそれ以上分割されないことを示す。ブロック1240が実行された後、サブルーチンはブロック1299から図24のブロック1200Bに戻り、領域Q4-H1及び領域Q4-H2に対する座標(領域1及び領域2に対する値を使用する)を出力パラメータとして戻す。後にわかるように、領域Q4-H2は一度も使用されず、従ってこの領域が0の幅を有していても構わない。

【0083】再び図24を参照すると、図25～図27のセグメント分割サブルーチン1200は次に、ブロック1200Bにおいてブロック1130によって2回目の呼び出しを受ける。以下の入力パラメータがサブルーチンに渡される。すなわち、領域Q4-H1の座標(我々の例においては155、76、200、0—図7参照)、Q1wt(70)及びQ4wt(50)である。出力パラメータの領域Q4-Q1及び領域Q4-Q2もサブルーチン1200に渡される。サブルーチン1200は領域Q4-H1を領域Q4-Q1及び領域Q4-Q2に分割し、これらの二つの領域の上部左及び下部右の角のx及びy座標を戻す。

【0084】領域Q4-H1を水平方向に分割するために、制御のフローは以前のようにブロック1215、1220及び1230を通してブロック1240に移る。ブロック1240は、領域1の下部右(LR)のy座標と領域2の上部左(UL)のy座標とを、分割すべきセグメントのULy座標から、セグメントの高さに $Wt1 / (Wt1 + Wt2)$ を乗じたものを減じたものに等しく設定する。この動作は図6に示されている。ブロック1240は、領域1のLRy座標を0(ブロック1205において初期化された)から以下のように変更することによって、セグメント(領域Q4-H1)を分割する。すなわち、領域1のLRy $=76 - (76 \times 120 / (70 + 50)) = 32$ である。領域2のULy座標も32に変更される。ブロック1240が実行された後、サブルーチンはブロック1299から図24のブロック1200Cに戻り、領域Q4-Q1及び領域Q4-Q2に対する座標(領域1及び領域2に対する値を使用する)を出力パラメータとして戻す。

【0085】再び図24を参照すると、図25～図27のセグメント分割サブルーチン1200は次に、ブロック1200Cにおいてブロック1130によって3回目の呼び出しを受ける。以下の入力パラメータがサブルーチンに渡される。すなわち、領域Q4-H2の座標(我々の例においては155、0、200、0)、Q3wt(0)及びQ4wt(0)である。出力パラメータの領域Q4-Q3もサブルーチン1200に渡される。しかし、サブルーチン1200が我々の重なった幅0の領域Q4-H2を分割する機会を有する前に、ブロック1210はWt1及びWt2を加え合わせ、それらが0以下であることを決定する。これは、ウィンドウに対してすでに十分な領域があることを示し、サブルーチンはブロック1298から図24のブロック1115に戻る。これらの二つの領域の上部左及び下部右の角のx及びy座標に対する初期化された値は戻され、それによって領域Q4-H2が分割されなかったことを示す。

【0086】再び図24を参照すると、ブロック1115は、最初の三つのウィンドウに対して計算された領域に対する座標をウィンドウ・データ40の領域44に書き込む。我々の例においては、次に最も高い活動状態にあるウィンドウ(WordPerfect)に対する領域44は、領域Q4-Q1の座標155、76、200、32でうめられる。次の最も高い活動状態にあるウィンドウ(Lotus 1-2-3)に対する領域44は、領域Q4-Q2の座標155、32、200、0でうめられる。これが最後のウィンドウであるので、ブロック1115の実行はここで停止する。

【0087】ブロック1120は、処理する必要があるウィンドウがまだ存在するか否かについて質問する。これは、それに対して領域が生成されなかった、サブルーチン1100に渡されたウィンドウ・リストにウィンドウが残されているか否かを見るために検査を行うことによって行われる。我々の例において我々はサブルーチン1100に渡された我々のウィンドウ・リストに二つのウィンドウを有していたため及びこれらのウィンドウの両方に対して領域が生成されたため、ブロック1120は否定的に答えられる。ブロック1130はいわゆる再帰的に呼び出し可能なサブルーチン1100であるので、サブルーチンはブロック1149においてブロック1130からブロック1139に戻る。ブロック1139は図23におけるブロック1150に戻る。

【0088】図23を参照すると、ブロック1150及び1155は、ウィンドウ・データ40におけるそれぞれのウィンドウに対する領域44において図24～図27のフロー・チャートによって提供されるデータに従って、ディスプレイ・スクリーン上にウィンドウを位置決めする。領域44に「ICON」を有する任意のウィンドウに対して、アイコンが生成される(Presentation Managerを使用して)。図8は我々の例のウィンドウがど

のようにユーザに表示されるかを示す。全てのウィンドウが処理されたことをブロック1150が示した時、ブロック1160は分類されたウィンドウ・リストにおける最初のウィンドウ（最も高い活動状態にあるウィンドウ）にフォーカスを与える。プログラムは次に、ユーザがディスプレイ・ウィンドウ・タイル表示モードを選択するか否かを再び見るために、ループをブロック401に戻る。

【0089】次に、まだ説明していない三つの他の実施例を説明するために、図24～図27に戻る。分割すべきセグメントがそれ自身、より大きなセグメントから垂直方向または水平方向に分割されたか否かを示すために、セグメント分割サブルーチン1200に付加的なパラメータが渡されることを除いて、図24は以前に説明したと同様に動作する。「水平」パラメータはサブルーチンが呼び出された時に初めて渡される。もし納税者タミーが比例するパラメータを得ることができたが、最適化パラメータは得ることができなかったならば、彼女はこの発明の第1の他の実施例を使用しており、ディスプレイ・スクリーン17は図9に示されるように見える。この実施例においては、最適化パラメータが得られないので、ブロック1220は否定的に答えられる。ブロック1245はスイッチとして働き、セグメントが分割（それによって垂直方向の分割を実行する）される最初の時にブロック1235を実行させ、セグメントがさらに分割される次の時にブロック1240を実行させ、それによって水平方向の分割を実行する。タイル表示領域の領域H1及び領域H2への分割は垂直方向に行われる。領域H1の領域Q1及び領域Q2への分割は水平方向に行われる。領域H2の領域Q3及び領域Q4への分割は水平方向に行われる。必要に応じて、領域Q4のさらなる分割が垂直方向に行われ、次に水平方向に行われる、等々される。サブルーチンは、それがブロック1299において呼び出された所に戻る。セグメントがどのように分割されたかに関する情報も、ブロック1299において戻される。

【0090】もしタミーが比例するパラメータを得ることができなかったが、最適化パラメータは得ることができたならば、彼女はこの発明の第2の他の実施例を使用しており、ディスプレイ・スクリーン17は図10に示されるように見える。この実施例においては、ブロック1215は否定的に答えられる。ブロック1250は、 $Wt2=0$ であるか否かを決定する。もしこれが真であれば、領域を見つけるためのウィンドウは一つしかなく、従ってセグメントは分割されない。その代わり、ブロック1251は、領域1の座標をブロック1205において初期化された同一の値に保つが、この領域が存在しないことを示すために領域2の座標を0、0、0、0に変更する。サブルーチンは、それがブロック1297において呼び出された所に戻る。もしブロック125

0が否定的に答えられれば、ブロック1255は最適化パラメータが得られるか否かについて質問する。この実施例においてはそれは得られるので、ブロック1255は肯定的に答えられる。ブロック1260は以前に述べたブロック1230と同様に働き、もしその幅が高さ以上であれば、セグメントを垂直方向に分割し、そうでなければセグメントを水平方向に分割する。ブロック1265は、以前に述べたブロック1235とほとんど同様に動作するが、ブロック1265はセグメントをいつも等分割する。同様に、ブロック1270は以前に述べたブロック1240とほとんど同様に動作するが、ブロック1275はセグメントをいつも等分割する。ブロック1265または1270が実行された後、サブルーチンはそれがブロック1296において呼び出された所に戻る。セグメントがどのように分割されたかに関する情報もブロック1296において戻される。

【0091】もしタミーが比例するパラメータを得ることができず、最適化パラメータも得ることができなかったならば、彼女はこの発明の第3の他の実施例を使用しており、ディスプレイ・スクリーン17は図11に示されるように見える。図11は図10と同一に見えるが、これはしばしば正しくない。この実施例においては、ブロック1255は否定的に答えられる。ブロック1275は以前に述べたブロック1245と同様に動作する。

【0092】図16～図19、図22及び図23のフロー・チャートとともに処理装置12において独立に実行する図28について説明する。ブロック1001は、処理すべきウィンドウ・イベントがあるか否かを見るために検査を行う。これは、図16及び図17のブロック110によって行われるのと同じの検査である。もしブロック1001が処理すべきウィンドウ・イベントがないことを決定すれば、ブロック1003は全てのウィンドウがクローズされたか否かを見るために検査を行う。もしそうであれば、プログラムはブロック1099において終了する。もしそうでなければ、プログラムは、処理すべきウィンドウ・イベントがあるか否かを見るために再び検査を行うために、ループをブロック1001に戻る。いったんブロック1001が肯定的に答えられれば、ブロック1010はウィンドウ・イベントを得る。ブロック1020は、このウィンドウ・イベントがユーザの処置によって生じたものであるか否かについて質問する。もしそうでなければ、プログラムは、処理すべき他のウィンドウ・イベントを探すためにループをブロック1001に戻る。なお、図16～図19は実際にイベントを実行する一図28はこの発明のウィンドウ・タイミング機能にインパクトを与える特定のイベントを探すだけである。もしブロック1020が肯定的に答えられれば、ブロック1050は制御データ30の中断フラグ34がTRUE2に等しいか否かを見るために検査を行う。もしこのフラグがTRUE2に等しければ、ウィン

ドウ・タイミング機能は非活動タイム・アウトにより中断された。ユーザは今や何かを行ったので、ウィンドウ・タイミング機能を再スタートさせるのが適当である。これは、ブロック1055において、図22のブロック205が肯定的に答えられ、ウィンドウ・データ30におけるタイマ・フラグ32がブロック208によって増加されるように中断フラグ34の値をFALSEに変更することによって行われる。

【0093】再び図28を参照すると、ブロック1050がどのように答えられるかとは無関係に、制御のフローはブロック1060に移り、ここで最後のイベント・フラグ36が制御データ30におけるタイマ32の値に設定される。従って、最後のイベント・フラグ36は、ディスプレイ・スクリーン上のウィンドウの一つに対するユーザの活動を示すイベントが生じた最後の時間を含む。この情報は、指定された非活動タイム・アウトが越えられたか否かを見るための検査を行うために、図16及び図17のブロック115及び118によって使用される。制御のフローはループをブロック1001に戻る。

【0094】以上、この発明を好ましい実施例及びいくつかの他の実施例について説明したが、当業者ならば、この発明の思想、範囲及び教示から逸脱することなく、詳細における種々の変更を行うことができるであろう。例えば、最も高い活動状態にあるウィンドウの位置は、図25～図27のブロック1235、1240、1265及び1270に多少変更を加えるだけで、スクリーンの左の代わりに、スクリーンの右側にすることができる。このため、何が最も高い活動状態にあるウィンドウに対する最も区別的な表示であると考えられるかについての文化的または個人的な相違を考慮に入れることができる。従って、ここに開示されたものは、特許請求の範囲において規定されたように限定されるだけである。

【0095】

【発明の効果】この発明は、コンピュータのディスプレイ・スクリーン上にウィンドウを区別的及び比例的に表示するための方法及び装置を提供する。

【図面の簡単な説明】

【図1】この発明のコンピュータ・システムのブロック図である。

【図2】ユーザがそのコンピュータ上で数分間または数時間タスクを実行した後、どのようにウィンドウが表示されるかを示す略線図である。

【図3】ディスプレイ・スクリーンの行及び列をどのようにx及びy座標にマップすることができるかを示す略線図である。

【図4】ディスプレイ・スクリーンのウィンドウ・タイル表示領域がどのようにこの発明による領域に分割されるかを示す略線図である。

【図5】ディスプレイ・スクリーンのウィンドウ・タイル表示領域がどのようにこの発明による領域に分割されるかを示す略線図である。

【図6】ディスプレイ・スクリーンのウィンドウ・タイル表示領域がどのようにこの発明による領域に分割されるかを示す略線図である。

【図7】ディスプレイ・スクリーンのウィンドウ・タイル表示領域がどのようにこの発明による領域に分割されるかを示す略線図である。

【図8】この発明の好ましい実施例においてどのようにウィンドウが比例的に表示されるかを示す略線図である。

【図9】この発明の第1の他の実施例においてどのようにウィンドウが比例的に表示されるかを示す略線図である。

【図10】この発明の第2の他の実施例においてどのようにウィンドウが区別的にしかし非比例的に表示されるかを示す略線図である。

【図11】この発明の第3の他の実施例においてどのようにウィンドウが区別的にしかし非比例的に表示されるかを示す略線図である。

【図12】この発明の制御データを示す略線図である。

【図13】分類される前のこの発明のウィンドウ・データを示す略線図である。

【図14】分類された後のこの発明のウィンドウ・データを示す略線図である。

【図15】この発明のユーザが変更可能なパラメータを設定するために使用されるスクリーンの一例を示す略線図である。

【図16】この発明のフロー・チャートである。

【図17】この発明のフロー・チャートである。

【図18】この発明のフロー・チャートである。

【図19】この発明のフロー・チャートである。

【図20】この発明のフロー・チャートである。

【図21】この発明のフロー・チャートである。

【図22】この発明のフロー・チャートである。

【図23】この発明のフロー・チャートである。

【図24】この発明のフロー・チャートである。

【図25】この発明のフロー・チャートである。

【図26】この発明のフロー・チャートである。

【図27】この発明のフロー・チャートである。

【図28】この発明のフロー・チャートである。

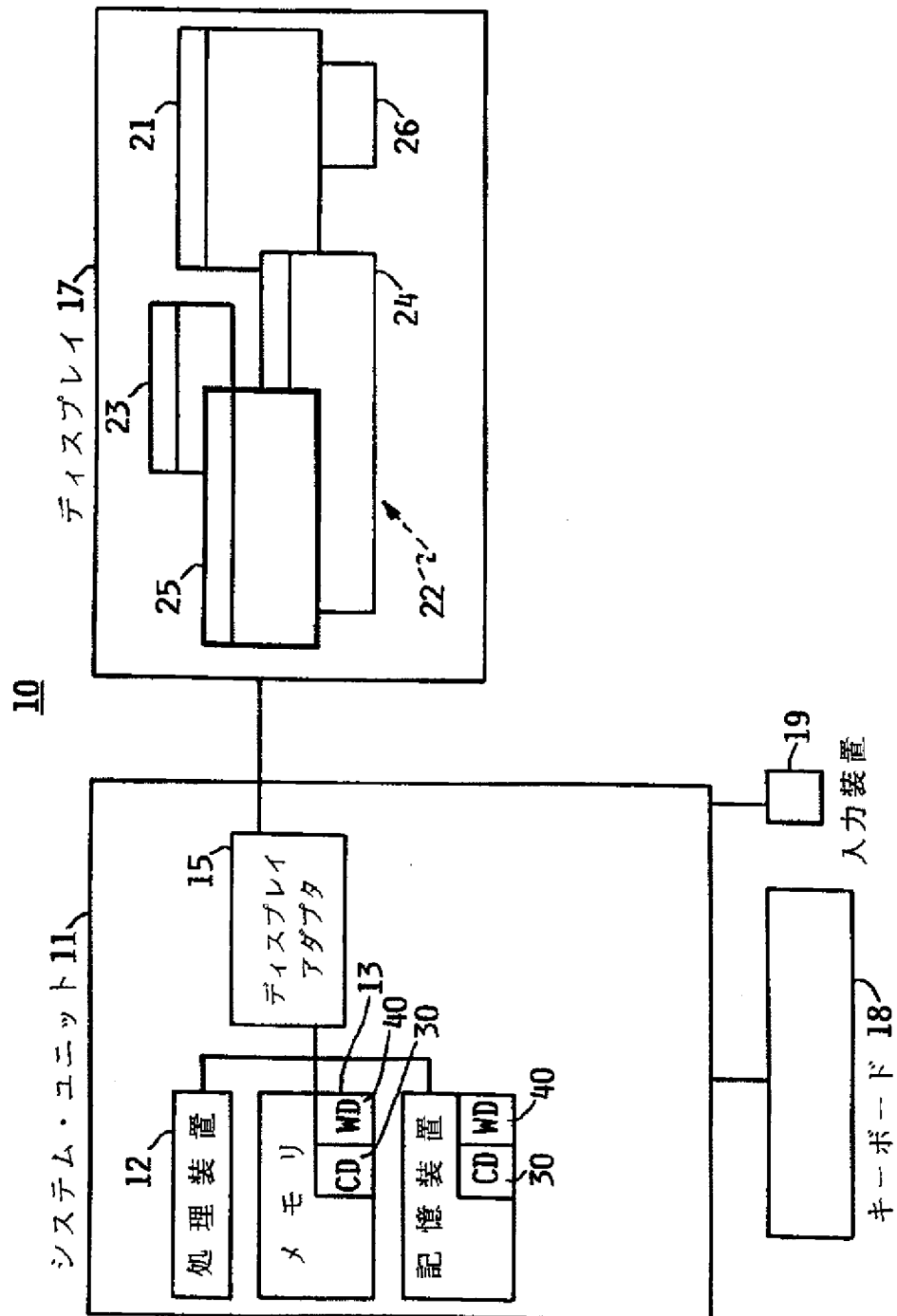
【符号の説明】

- 10 コンピュータ・システム
- 11 システム・ユニット
- 12 処理装置
- 13 メモリ
- 14 記憶装置
- 15 ディスプレイ・アダプタ
- 17 ディスプレイ

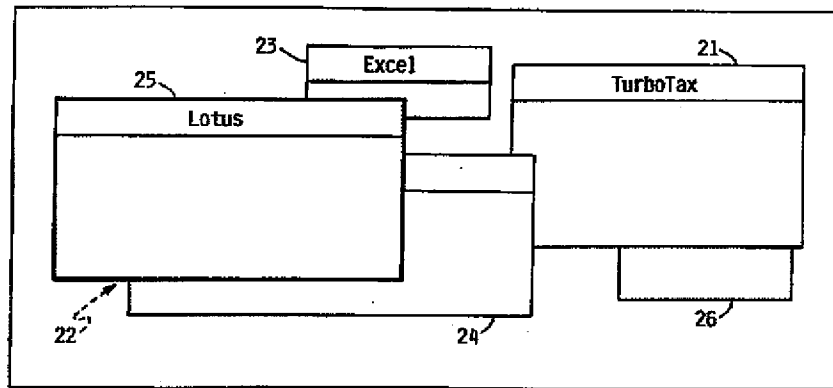
18 キーボード
19 入力装置
21~26 ウィンドウ

30 制御データ
40 ウィンドウ・データ

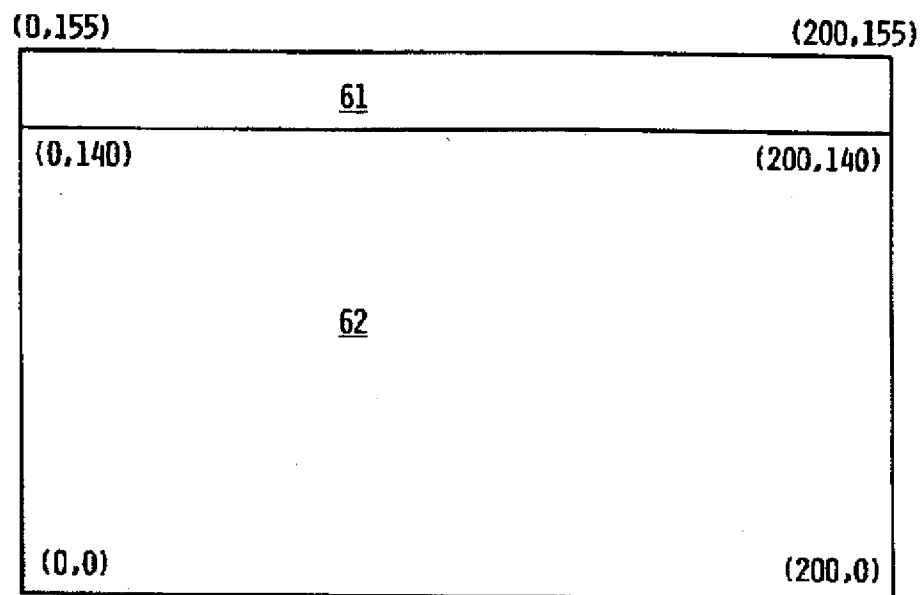
【図1】



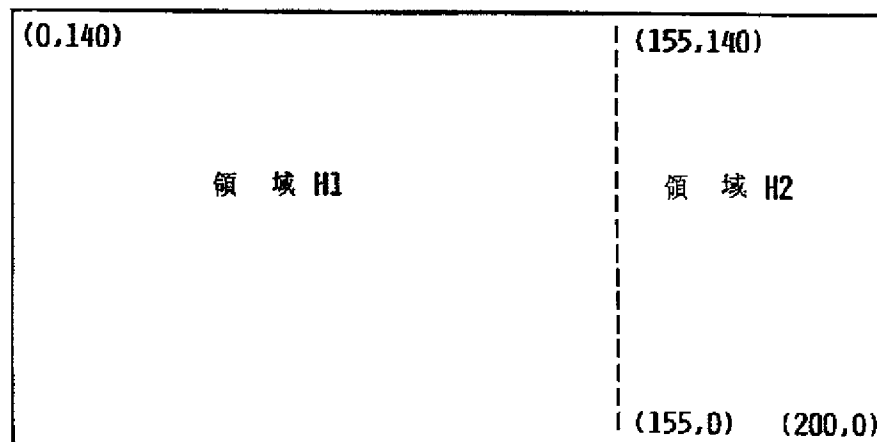
【図2】



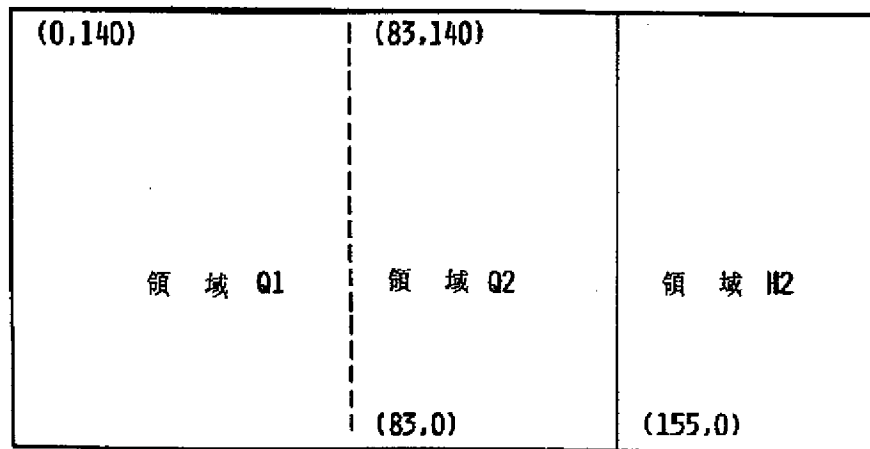
【図3】



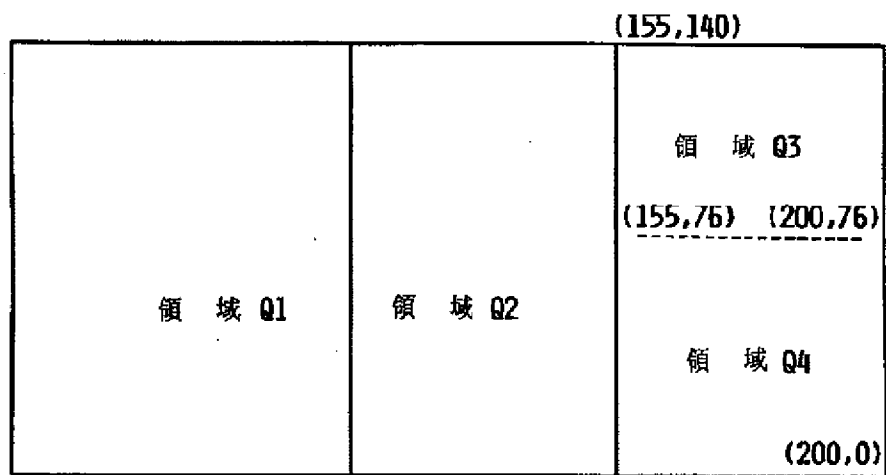
【図4】



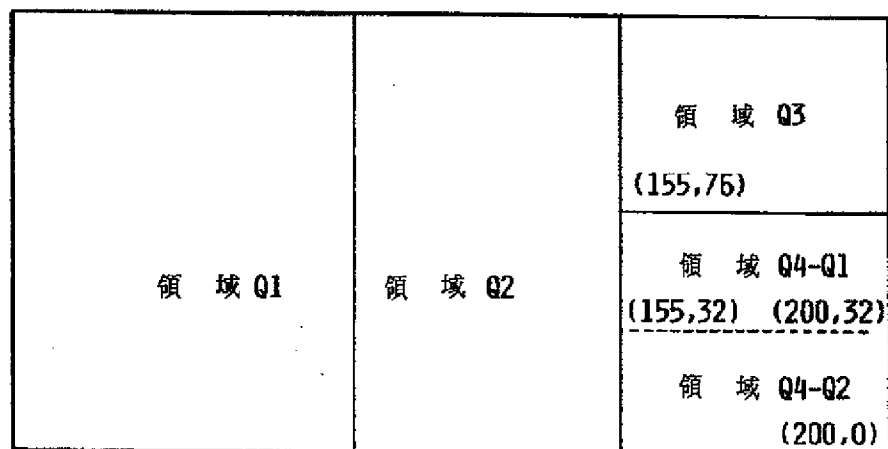
【図 5】



【図 6】



【図 7】



【図8】

FM			
Turbo Tax	Quicken	Excel	
		Word Perfect MY.D	
		Lotus 1-2-3	

【図9】

FM			
Turbo Tax		Excel	
		Word Perfe	Lotus
Quicken			

【図10】

FM			
Turbo Tax		Excel	
Quicken		Word Perfect	Lotus 1-2-3

【図11】

FM			
Turbo Tax		Excel	
Quicken		Word Perfect	Lotus 1-2-3

【図 1 2】

制 御 デ ー タ <u>30</u>	
31へオン/オフ・フラグ	ON
32へタイム	000001000
33へサンプリング・レート	2
34へ中 断	FALSE
36へ最後のイベント	996
37へ非活動タイム・アウト	5
38へ保 管	ON
51へタイトル表示領域 (ULx, y, LRx, y)	0, 140, 200, 0
52へ比例タイトル表示	ON
53へ最適化タイトル表示	ON
54へ最小タイトル表示パーセント	5

【図13】

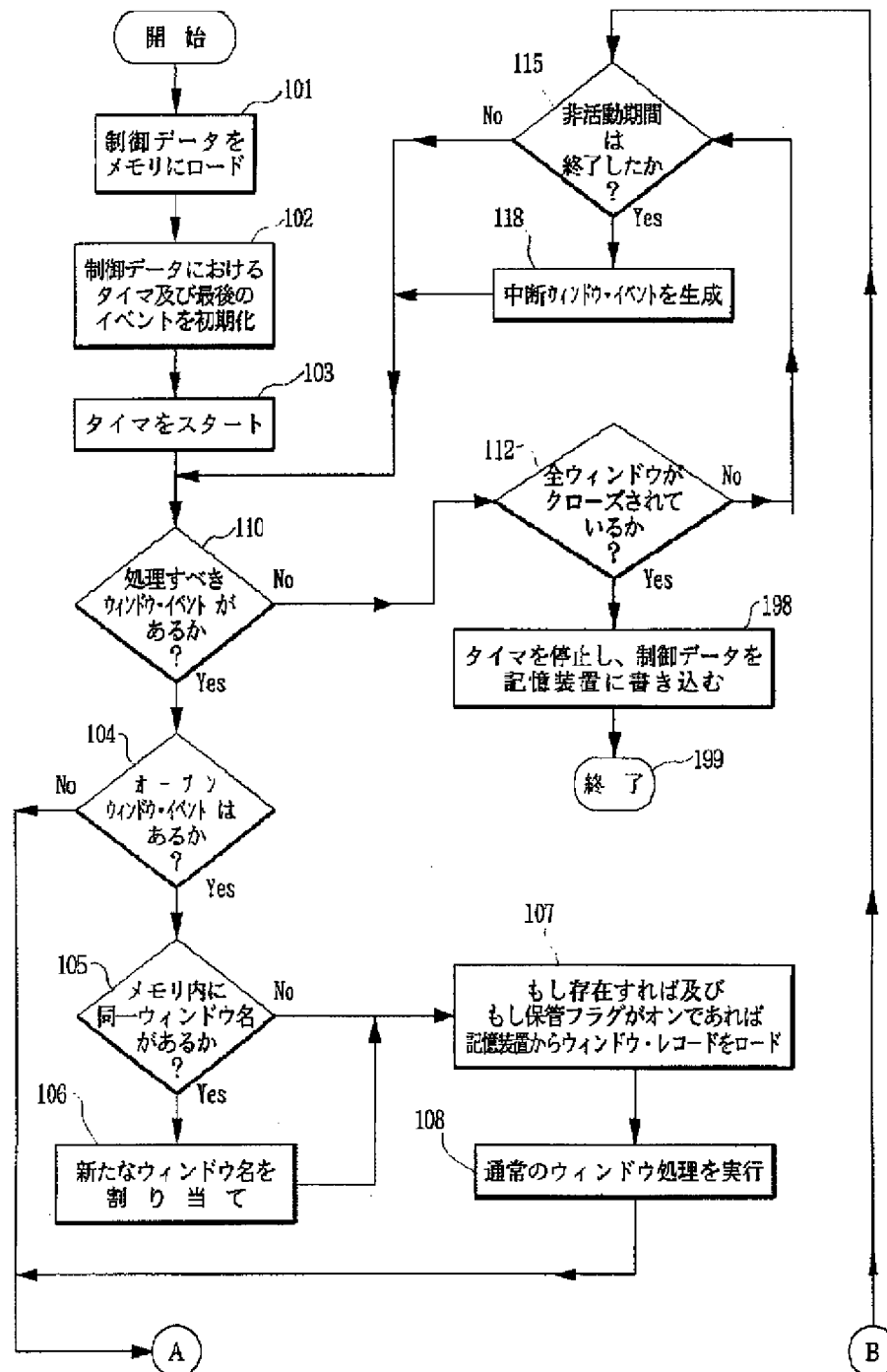
ウインドウ・データ 40			
41、 ウインドウ名	42、 イン・フォーカス	43、 合 計	44、 領 域 (ULX,Y,LRX,Y)
Word Perfect MY.TXT	900	70	0,0,0,0
File Manager	10	30	0,0,0,0
Lotus 1-2-3	985	50	0,0,0,0
Quicken	800	350	0,0,0,0
Excel	750	100	0,0,0,0
Turbo Tax	875	400	0,0,0,0

【図14】

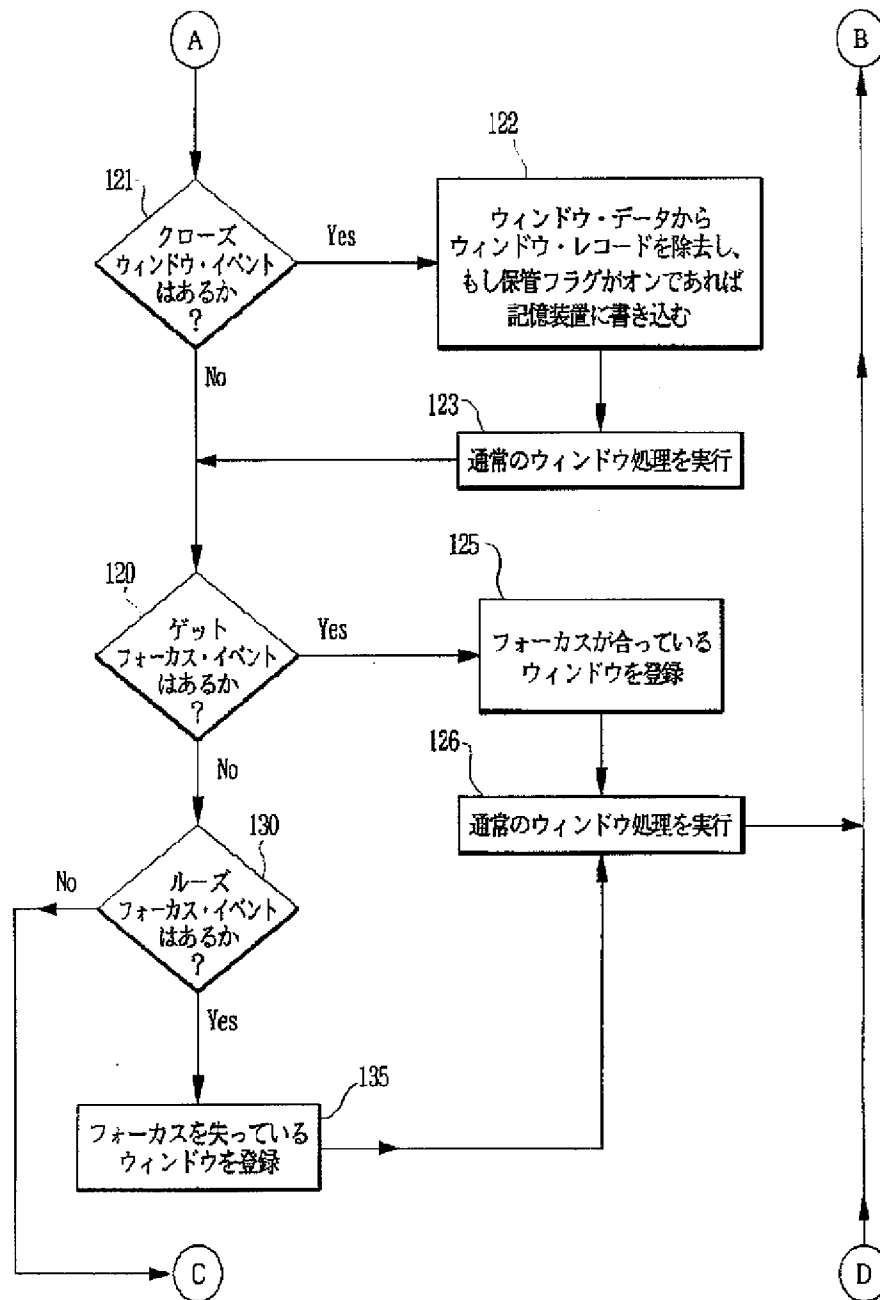
ウインドウ・データ (分類後) 40			
41、 ウインドウ名	42、 イン・フォーカス	43、 合 計	44、 領 域 (ULX,Y,LRX,Y)
Turbo Tax	875	400	0,140,83,0
Quicken	800	350	83,140,155,0
Excel	750	100	155,140,200,76
Word Perfect MY.TXT	900	70	155,76,200,32
Lotus 1-2-3	985	50	155,32,200,0
File Manager	10	30	ICON

Window Timing Function Parameters				
Window Timing Function	X	ON	OFF	
Sampling Rate		__30	seconds	
Inactivity Timeout		__5	minutes	
Suspend Timing?		YES	X	NO
Save Window Function Data?	X	YES		NO
Reset Window Timing?		YES	X	NO
Tiling Area	ULx	__0	ULy	140 LRx 200 LRY __0
Proportional Tiling	X	ON		OFF
Best Fit	X	ON		OFF
Minimum Tiling Percentage		__5	Percent	

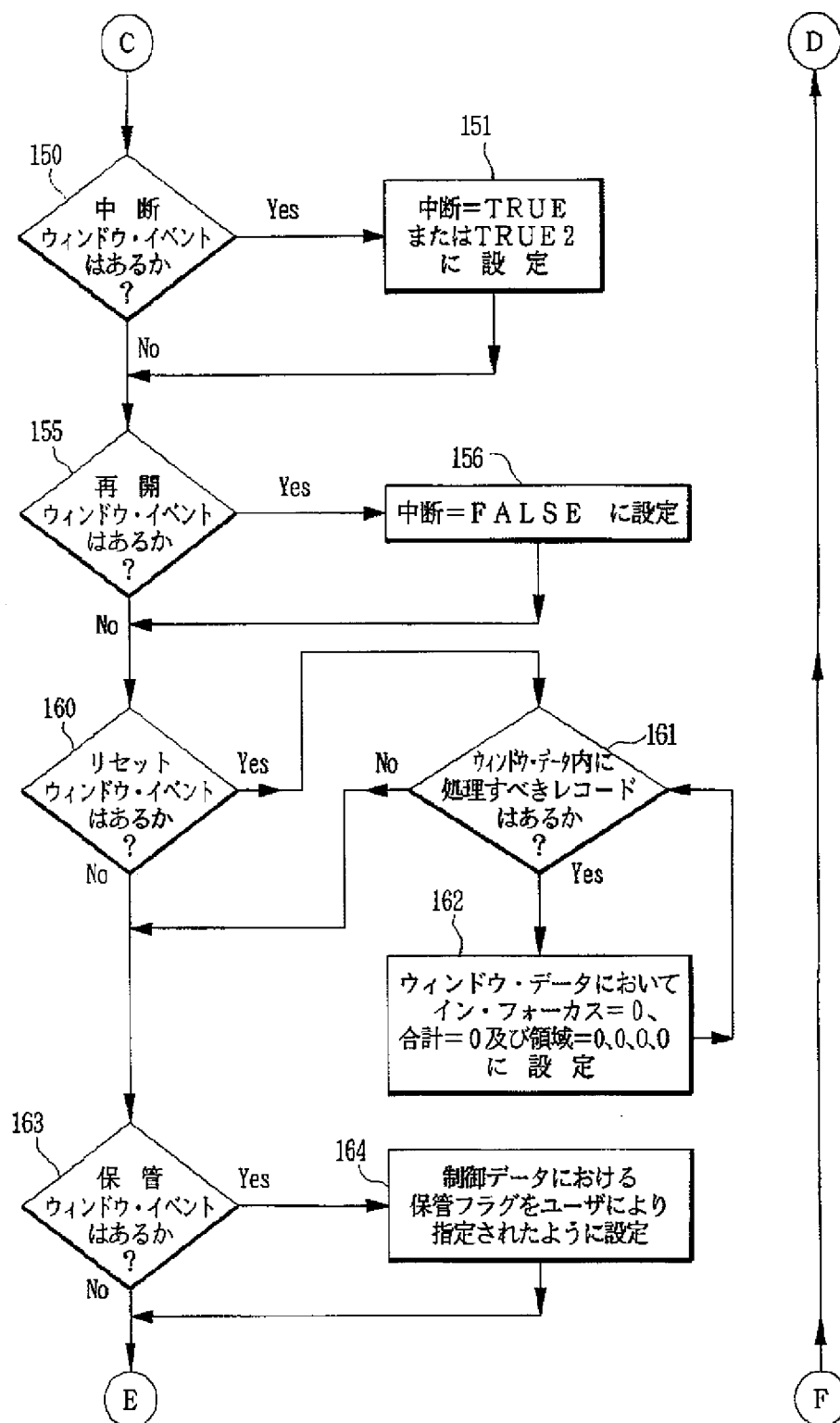
【図16】



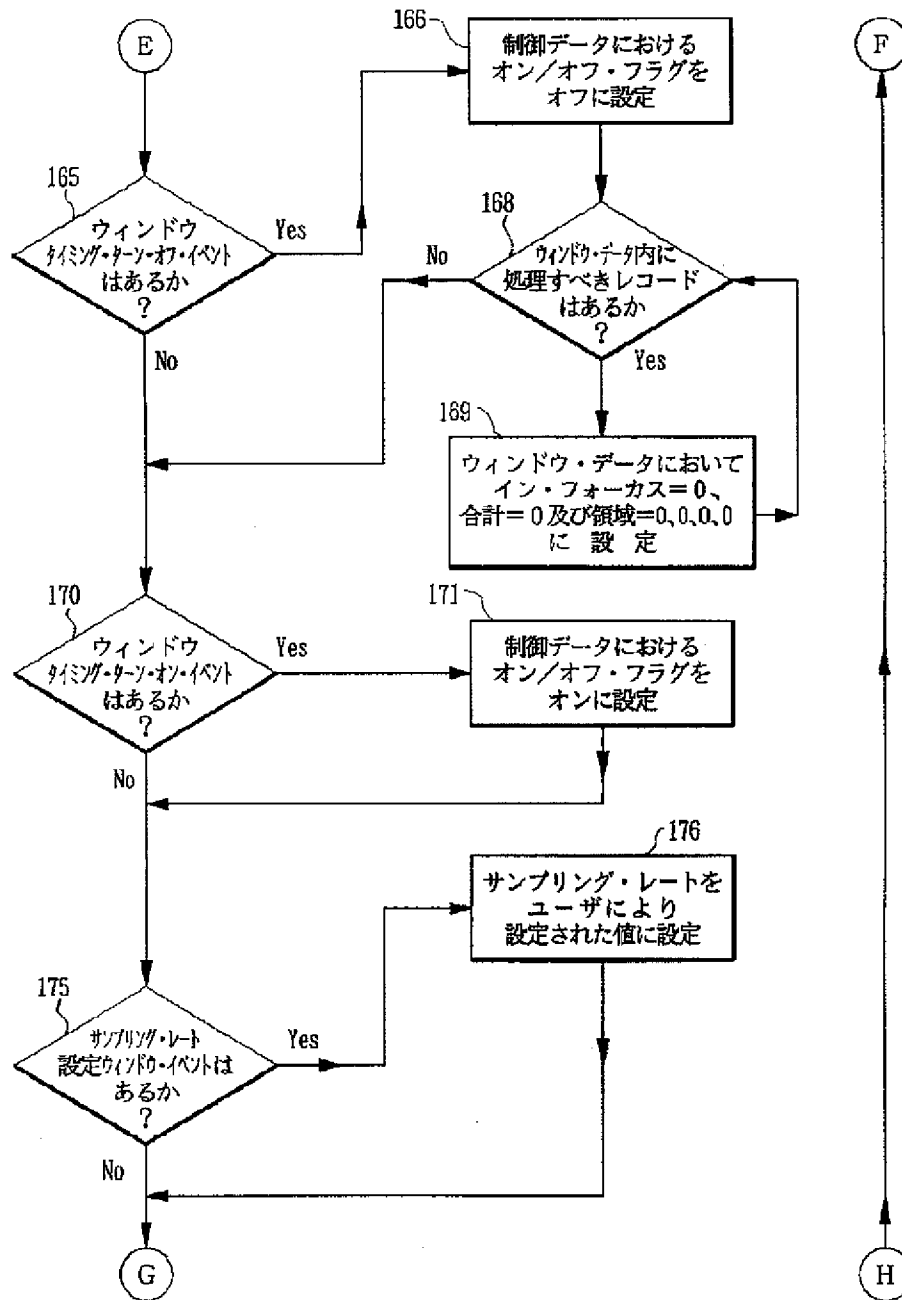
【図 17】



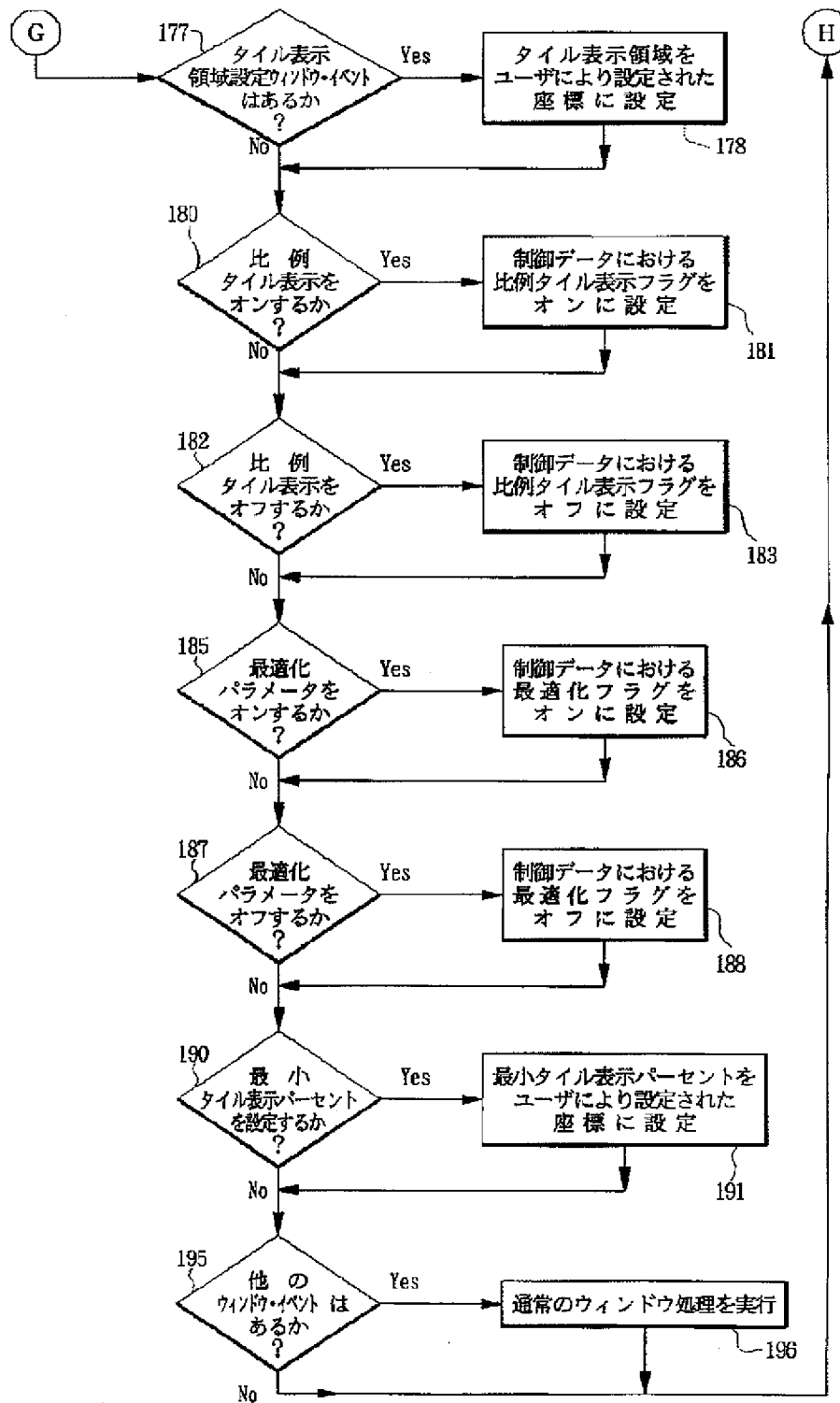
【図 18】



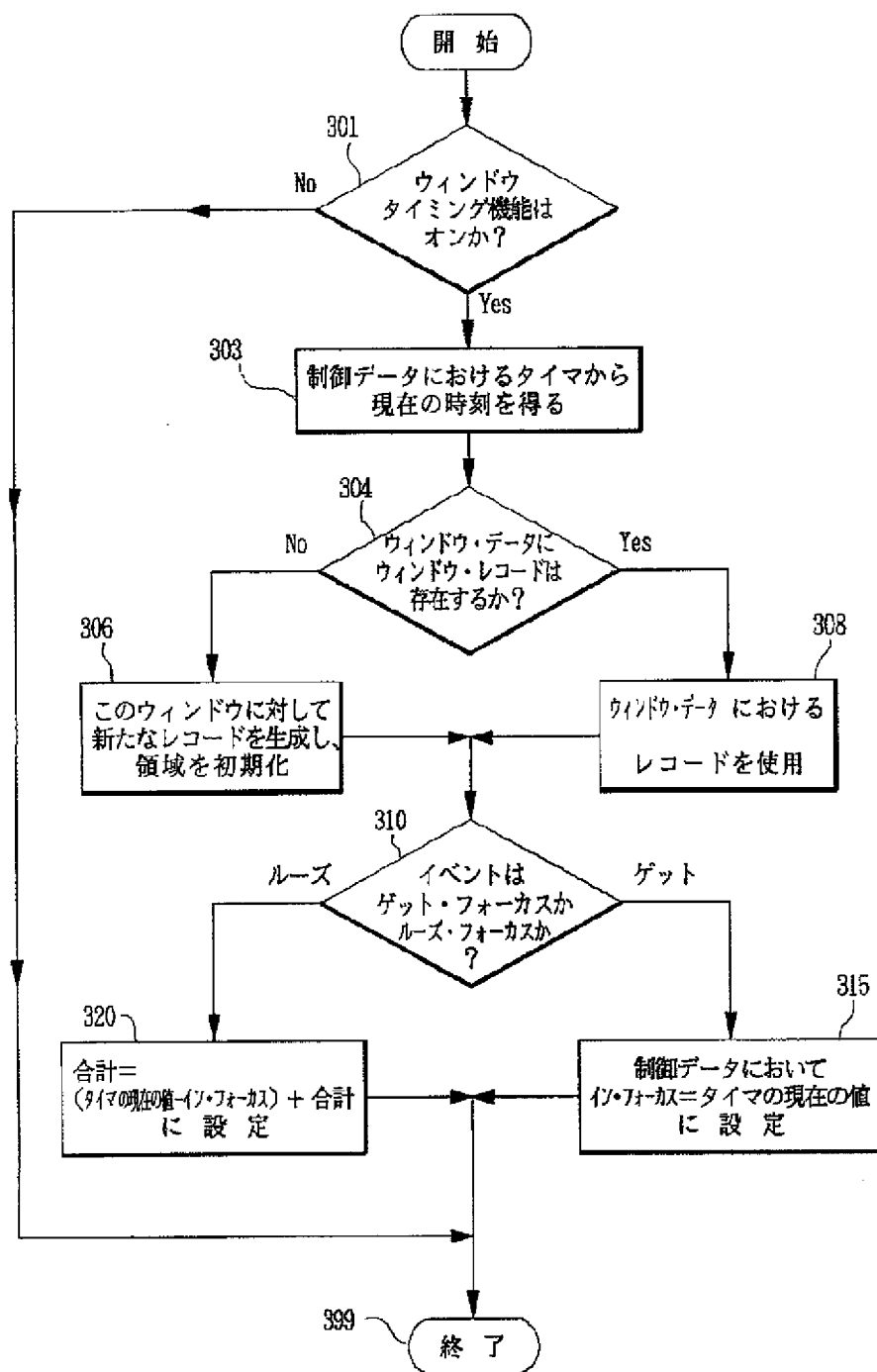
【図19】



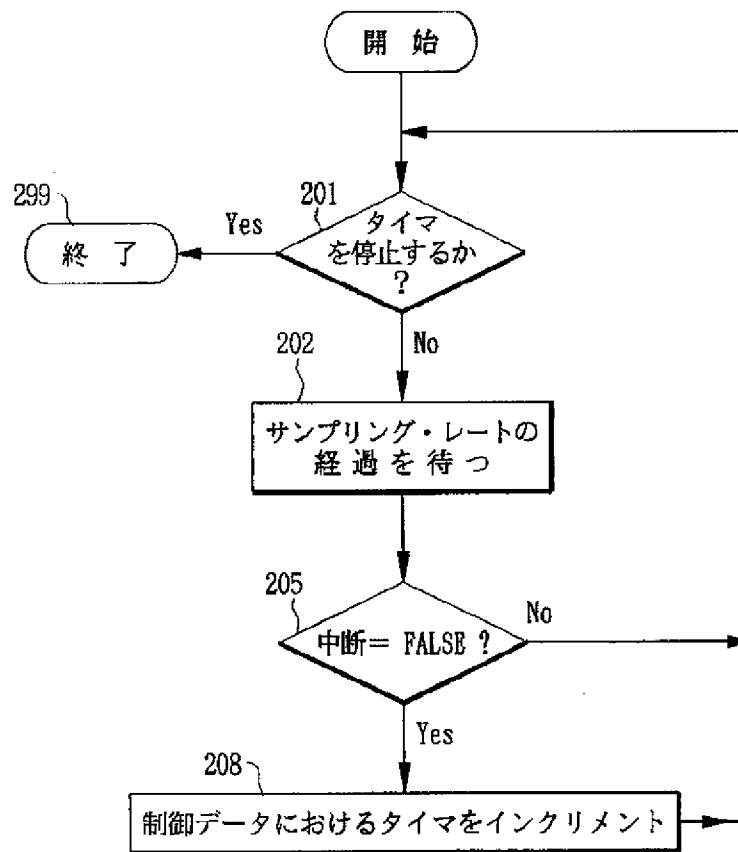
【図20】



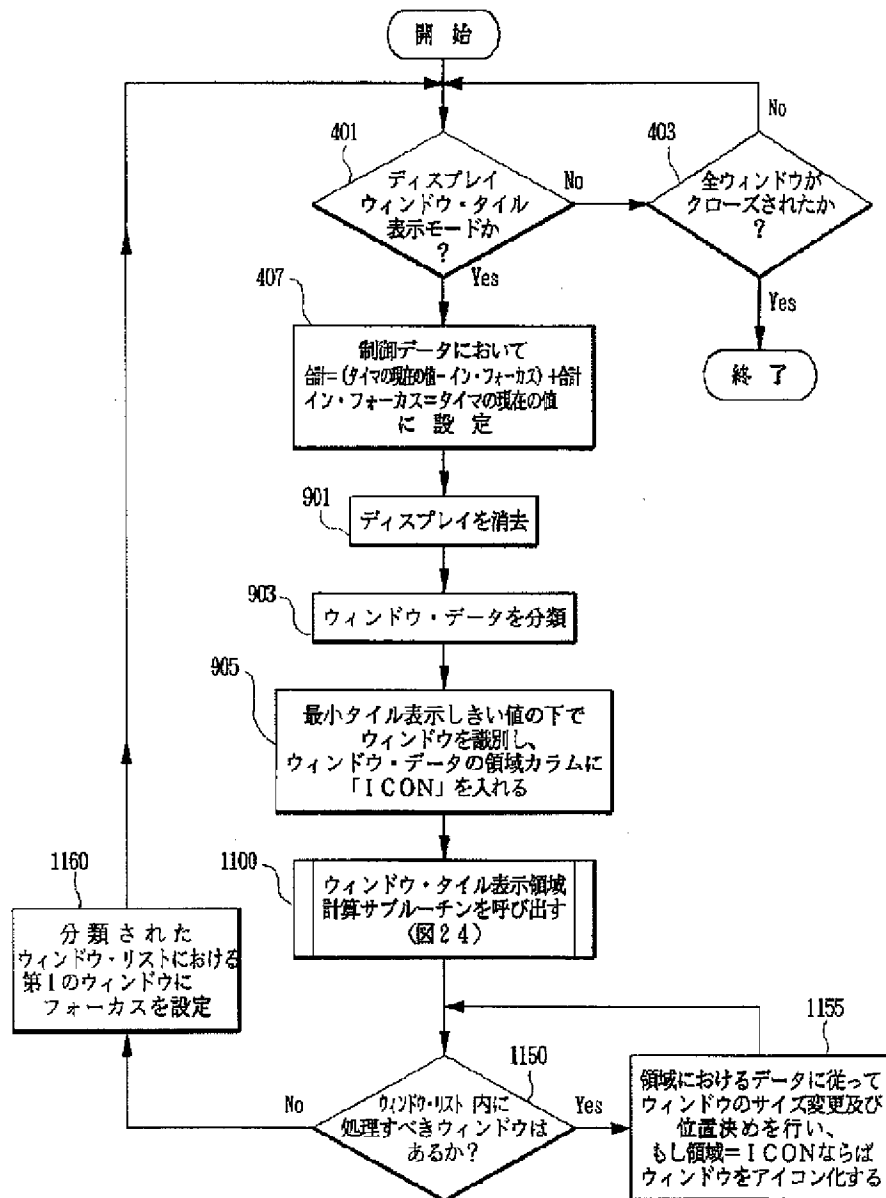
【図21】



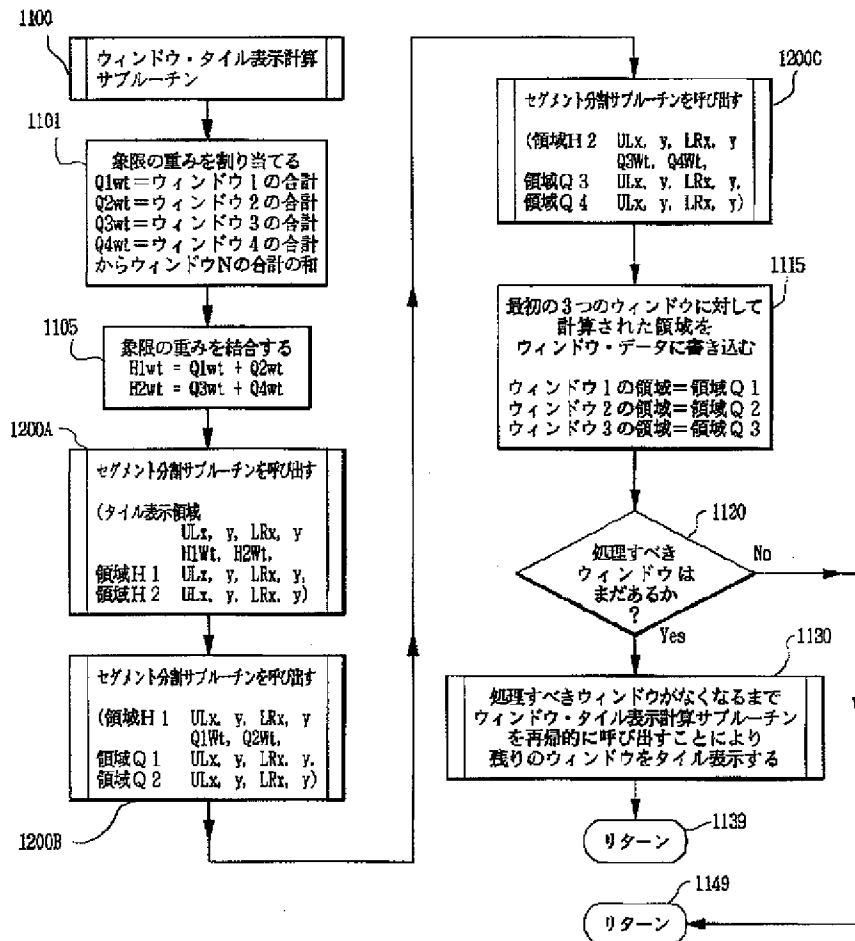
【図 22】



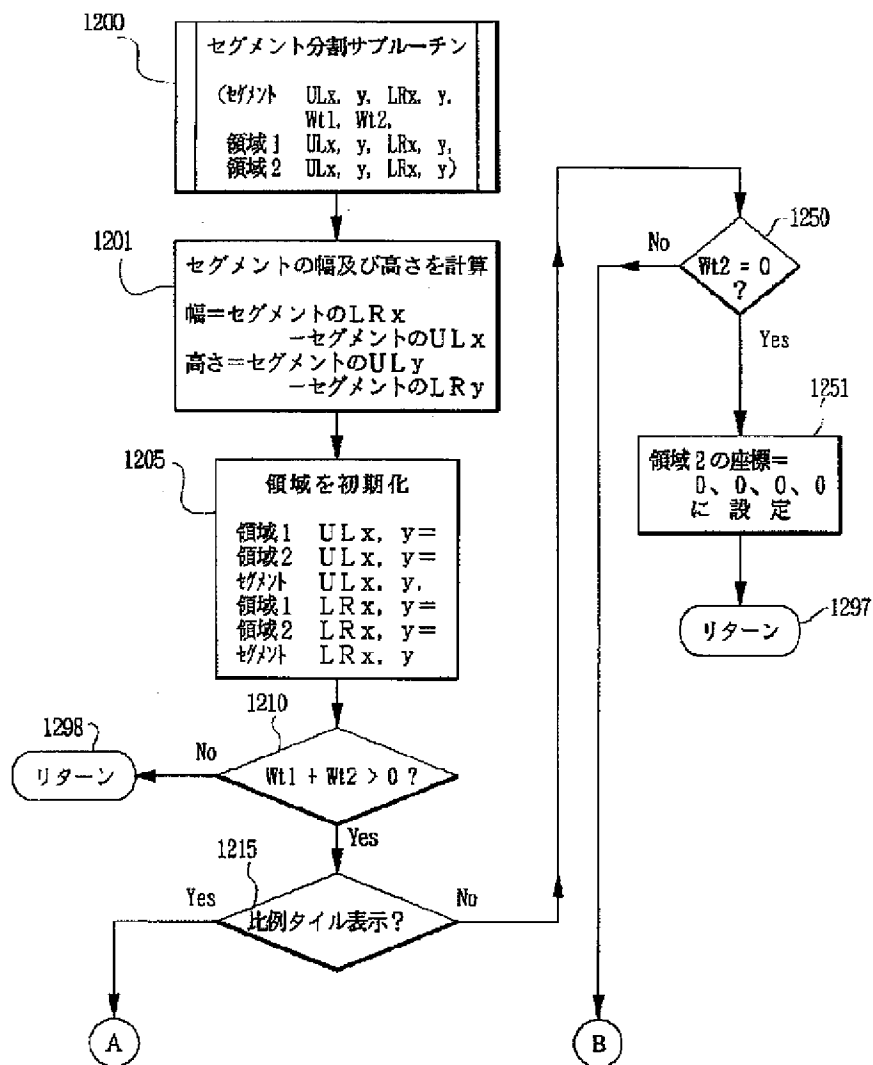
【図 23】



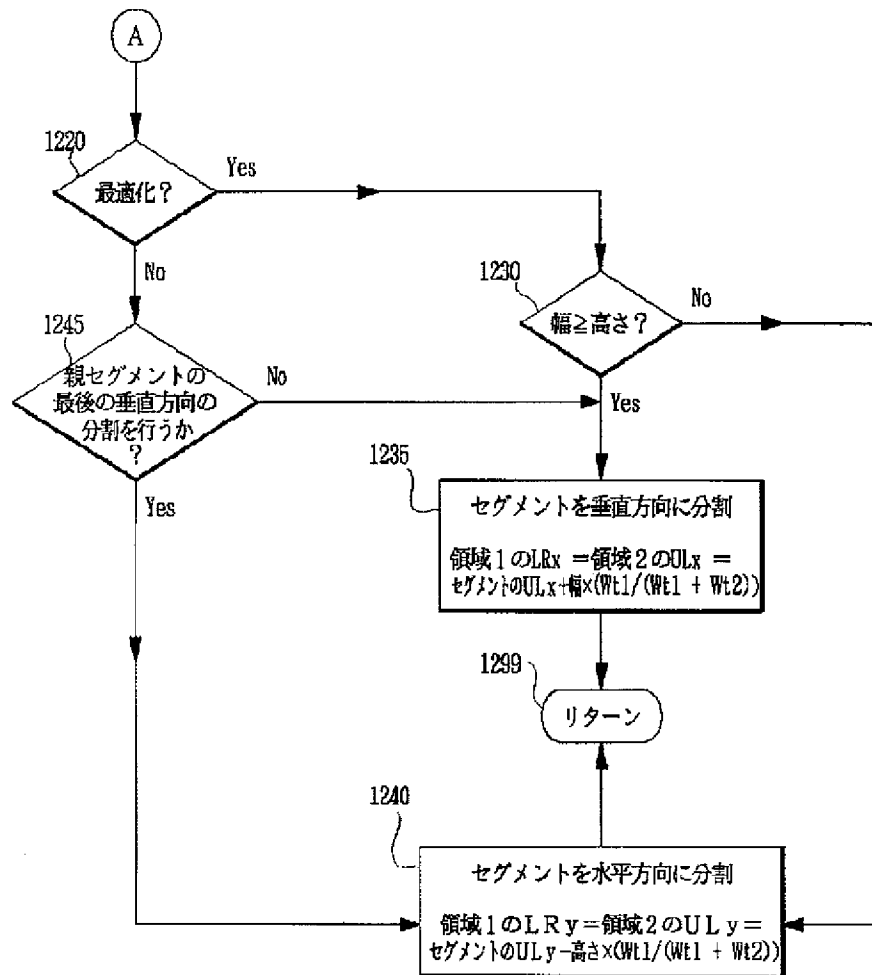
【図 24】



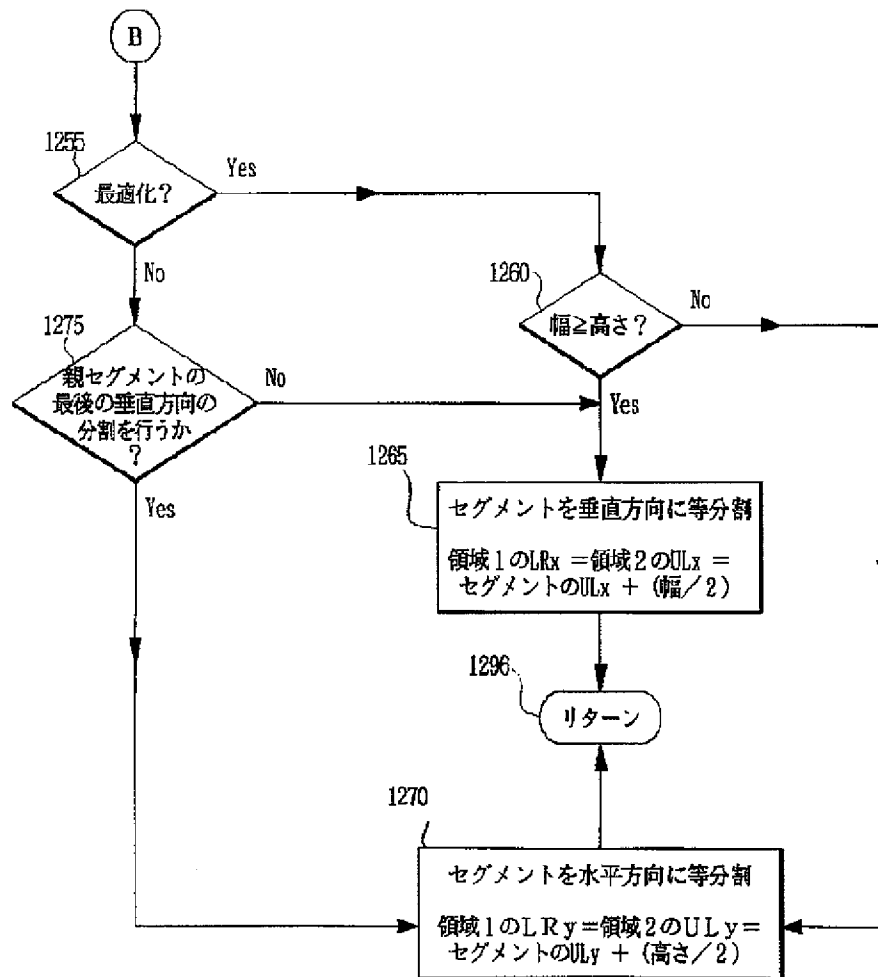
【図25】



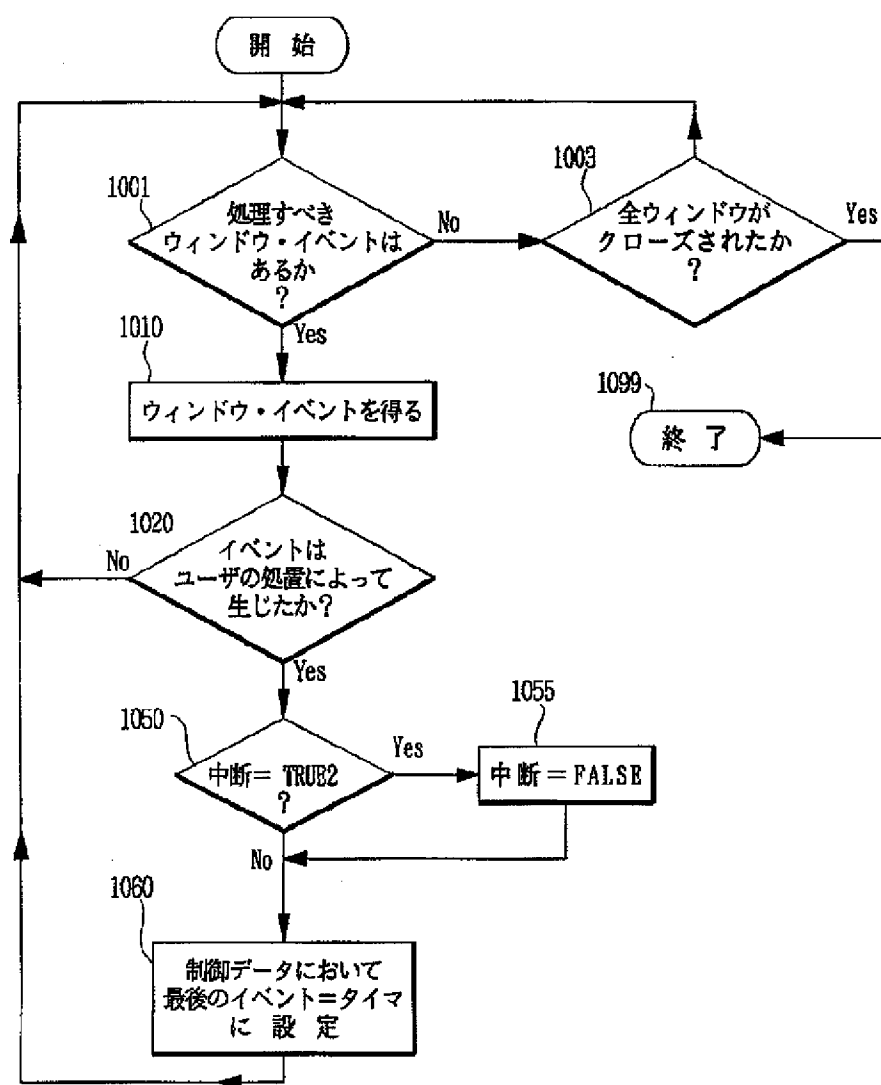
【図26】



【図27】



【図28】



フロントページの続き

(72) 発明者 ジェフリ・マイケル・ライアン
 アメリカ合衆国、ミネソタ州パイロン、セ
 カンドストリート ノースウエスト 6